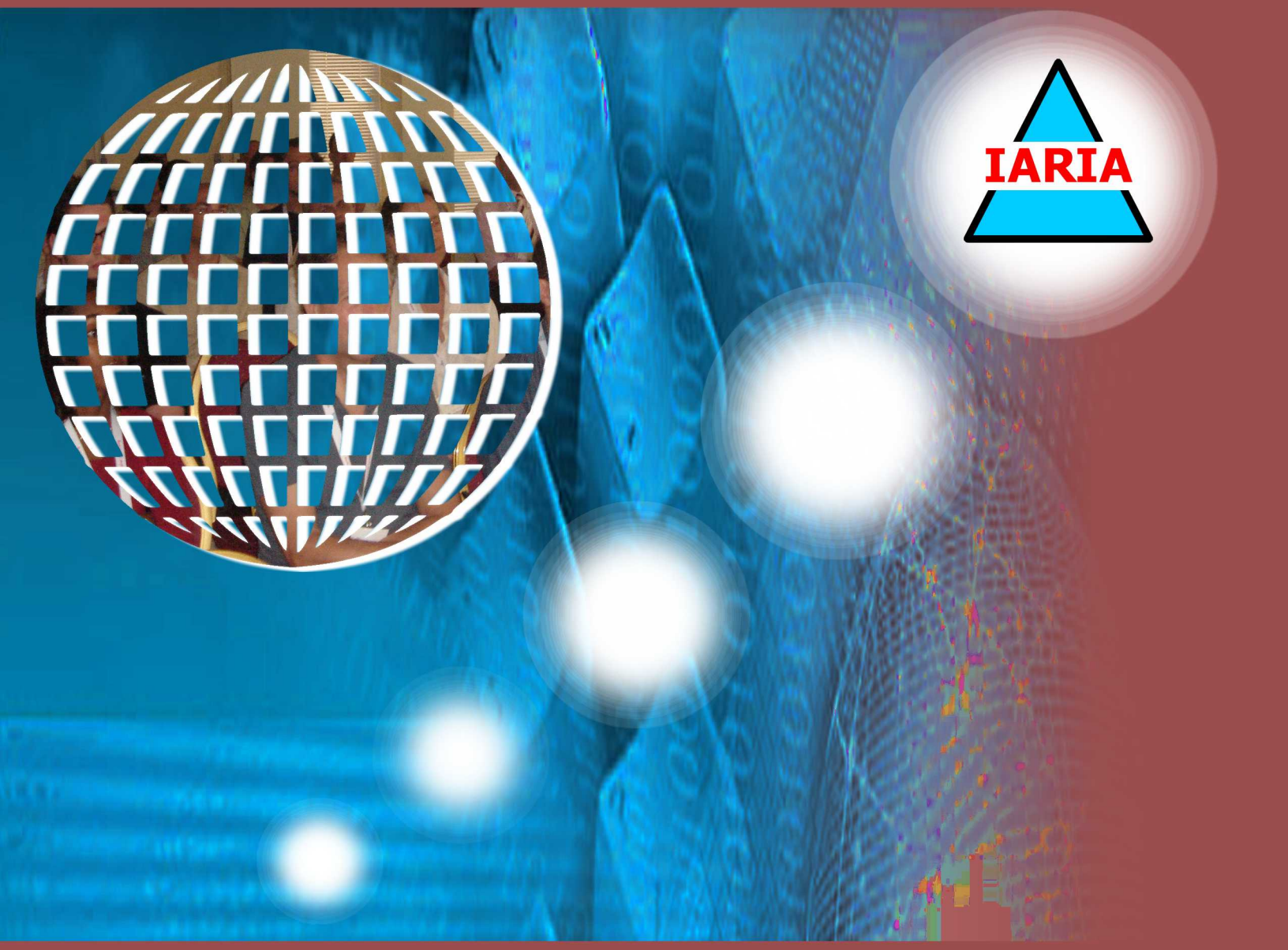
International Journal on Advances in Security



2026 vol. 19 nr. 1&2

The *International Journal on Advances in Security* is published by IARIA.
ISSN: 1942-2636
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

IARIA is the owner of the publication and of editorial aspects. IARIA reserves the right to update the content for quality improvements.

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy or print, providing the reference is mentioned and that the resulting material is made available at no cost.

Reference should mention:

International Journal on Advances in Security, issn 1942-2636
vol. 19, no. 1 & 2, year 2026, http://www.iariajournals.org/security/

The copyright for each included paper belongs to the authors. Republishing of same material, by authors or persons or organizations, is not allowed. Reprint rights can be granted by IARIA or by the authors, and must include proper reference.

Reference to an article in the journal is as follows:

<Author list>, "<Article title>"
International Journal on Advances in Security, issn 1942-2636
vol. 19, no. 1 & 2, year 2026, <start page>:<end page> , http://www.iariajournals.org/security/

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA
www.iaria.org

Copyright © 2026 IARIA

International Journal on Advances in Security
Volume 19, Number 1 & 2, 2026

Editors-in-Chief

Hans-Joachim Hof,
- Full Professor at Technische Hochschule Ingolstadt, Germany
- Lecturer at Munich University of Applied Sciences
- Group leader MuSe - Munich IT Security Research Group
- Group leader INSicherheit - Ingolstädter Forschungsgruppe angewandte IT-Sicherheit
- Chairman German Chapter of the ACM

Editorial Board

Oum-El-Kheir Aktouf, Univ. Grenoble Alpes | Grenoble INP, France
Eric Amankwa, Presbyterian University, Ghana
Ilija Basicevic, University of Novi Sad, Serbia
Cătălin Bîrjoveanu, "Al.I.Cuza" University of Iasi, Romania
Steve Chan, Decision Engineering Analysis Laboratory, USA
Abdullah S. Al-Alaj, Virginia Wesleyan University, USA
El-Sayed M. El-Alfy, King Fahd University of Petroleum and Minerals, Saudi Arabia
Rainer Falk, Siemens Technology, Germany
Steffen Fries, Siemens AG, Germany
Damjan Fujs, University of Ljubljana, Slovenia
Hans-Joachim Hof, Technische Hochschule Ingolstadt, Germany
Gahangir Hossain, University of North Texas, Denton, USA
Fu-Hau Hsu, National Central University, Taiwan
Sokratis Katsikas, Norwegian University of Science and Technology - NTNU, Norway
Hyunsung Kim, Kyungil University, Korea
Dragana Krstic, University of Nis, Serbia
Yosra Lakhdhar, Digital Research Center of Sfax (CRNS) / CN&S Research Lab at SUP'COM, Tunisia
Petra Leimich, Edinburgh Napier University, UK
Shimin Li, Winona State University, USA
Yi Liu, University of Massachusetts Dartmouth, USA
Giuseppe Loseto, LUM "Giuseppe Degennaro" University, Italy
Mohammadreza Mehrabian, South Dakota School of Mines and Technology, USA
Carla Merkle Westphall, Federal University of Santa Catarina (UFSC), Brazil
Aleksandra Mileva, Goce Delcev University, Republic of N. Macedonia
Vasudevan Nagendra, Sekyurity AI, USA
Brajendra Panda, University of Arkansas, USA
Paweł Rajba, University of Wroclaw, Poland
Danda B. Rawat, Howard University, USA
Claus-Peter Rückemann, Universität Münster / DIMF / Leibniz Universität Hannover, Germany
Antonio Ruiz Martínez, University of Murcia, Spain
Rocky Slavin, University of Texas at San Antonio, USA

Pedro Sousa, University of Minho, Braga, Portugal
Miroslav Velev, Aries Design Automation, USA
Cong-Cong Xing, Nicholls State University, USA

International Journal on Advances in Security
Volume 19, Numbers 1 & 2, 2026

CONTENTS

pages: 1 - 11
From Cockpit to Control Tower: Systematic Assessment of Cyber Threat Actors, Attack Vectors, and Vulnerabilities in Civil Aviation
Alexander Lawall, IU International University of Applied Sciences, Deutschland

pages: 12 - 21
Professional Experience May Matter More Than LLM Support when Selecting and Prioritizing Cybersecurity Requirements
Damjan Fujs, University of Ljubljana, Faculty of Computer and Information Science, Slovenia
Damjan Vavpotič, University of Ljubljana, Faculty of Computer and Information Science, Slovenia
Tomaž Hovelja, University of Ljubljana, Faculty of Computer and Information Science, Slovenia
Marko Poženel, University of Ljubljana, Faculty of Computer and Information Science, Slovenia

pages: 22 - 32
Obstacles in Implementation, Integration, and Security of Self-Sovereign Identity and Proposed Solutions
Erik Hieta-aho, VTT Technical Research Centre of Finland, Finland
Anni Karinsalo, VTT Technical Research Centre of Finland, Finland
Valtteri Lipiäinen, VTT Technical Research Centre of Finland, Finland

pages: 33 - 44
Near-Real-Time Fraud Detection in Banking Using HTAP Architecture: A Benchmarking-Driven Evaluation
Matteo G. Giorgino, University of Malta, Malta
Joseph G. Vella, University of Malta, Malta

pages: 45 - 54
Confirming the Trustworthiness of a Credential Issuer Using Remote Attestation
Rainer Falk, Siemens AG, Germany
Steffen Fries, Siemens AG, Germany

pages: 55 - 72
A Life-Cycle Model for Domain Security With Deterministic Multi-Criteria Relevance Scoring for Look-Alike Domains
Thomas Fritzler, Hochschule Darmstadt (h_da), Germany
Michael Massoth, Hochschule Darmstadt (h_da), Germany

pages: 73 - 86
Executive Cyber Risk Governance Using Integrated Key Performance and Key Risk Indicators: The BCCR-Card Framework
Alexander Lawall, IU International University of Applied Sciences, Deutschland
Maik Drozdzynski, IU International University of Applied Sciences, Deutschland

pages: 87 - 98
Practical Evaluation of Password-Authenticated Key Exchange Schemes in Classical and Post-Quantum Settings on Android

Jörn-Marc Schmidt, IU International University of Applied Sciences, Deutschland
Alexander Lawall, IU International University of Applied Sciences, Deutschland

pages: 99 - 109
Beyond Prevalence: Quantifying the Behavioural Effectiveness of Persuasion Principles in Phishing Attacks
Alexander Lawall, IU International University of Applied Sciences, Deutschland

pages: 110 - 119
Secure Software Brownfield Engineering
Aspen Olmsted, Wentworth Institute of Technology, United States

pages: 120 - 129
User-Centric Privacy in Smart Healthcare: Cross-Service Analysis and Design Implications
Elias Seid, Stockholm University, Sweden
Huan Guo, Stockholm University, Sweden
Fredrik Blix, Stockholm University, Sweden
Yuhong Li, Stockholm University, Sweden

pages: 130 - 142
Automated AWS Security Scanning and DORA Article Mapping: Evidence-Based Compliance for Financial Institutions
Elias Seid, Stockholm University, Sweden
Naol Mengistu, Stockholm University, Sweden
Ali Ferzali, Stockholm University, Sweden
Fredrik Blix, Stockholm University, Sweden

pages: 143 - 153
Isogeny-Based Identification Using Modified Schnorr Sigma Protocol
Mahdi Mahdavi, Universitat Oberta de Catalunya, UOC-Tech, Spain
Zaira Pindado, Barcelona Supercomputing Center (BSC), Spain
Helena Rifà-Pous, Universitat Oberta de Catalunya, UOC-Tech, Spain
Amineh Sakhaie, University of Lisbon, Portugal

pages: 154 - 170
Threat-Centric Vulnerability Management: Explainable Multi-Modal Learning for Mapping CVEs to the MITRE ATT&CK Framework
Logan McMahon, Queen's University Belfast, United Kingdom
Oluwafemi Olukoya, Queen's University Belfast, United Kingdom

pages: 171 - 198
MOTRA: Data Generation for HIDS & NIDS in OT-focused Environments
Peter Heller, OTH Regensburg, LaS³, Germany
Sebastian Kraust, OTH Regensburg, LaS³, Germany
Jürgen Mottok, OTH Regensburg, LaS³, Germany

pages: 199 - 207
Optimizing Certificate Validation in OT Environments by Caching Certificate Validation Results
Steffen Fries, Siemens AG, Germany
Rainer Falk, Siemens AG, Germany
Andreas Guettinger, Siemens AG, Germany

From Cockpit to Control Tower: Systematic Assessment of Cyber Threat Actors, Attack Vectors, and Vulnerabilities in Civil Aviation

Alexander Lawall
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
alexander.lawall@iu.org

***Abstract*—Civil aviation increasingly depends on interconnected digital systems spanning airborne platforms, ground infrastructure, and communication links, expanding the sector's cyber-attack surface. The paper synthesizes the civil aviation threat landscape via a literature-informed, interview study with five commercial airline pilots (A320/A330/340/CRJ-900). It maps twelve representative vectors to the International Civil Aviation Organization (ICAO) Doc 9859 probability-severity matrices guided by three research questions on threat actors, attack vectors, and system vulnerability. In the sample, nation-states, cybercriminals, and insiders were perceived as the most credible adversaries. Global Navigation Satellite Systems (GNSS) spoofing, Instrument Landing System (ILS) spoofing, and Avionics exploitation were rated as higher-risk scenarios, while weaknesses in ground Information Technology (IT) and unauthenticated protocols (e.g., Aircraft Communications Addressing and Reporting System (ACARS)) were frequently noted. The findings result in prioritized mitigations (signal authentication, architectural segregation, auditable update processes) and argue for enforceable regulatory baselines. The contribution is a practitioner-informed, ICAO-aligned risk classification that integrates pilot perceptions with literature evidence, offering an actionable bridge between cybersecurity and aviation safety management.**

Keywords-Civil Aviation; Cybersecurity; Threat Landscape; Vulnerability Assessment; Attack Vectors; Pilot-based Risk Classification; Critical Infrastructure.

I. Introduction

This work is an extended version of *Cybersecurity in Civil Aviation – Threat Landscape and Vulnerability Assessment of Attack Vectors*, published at IoTAI 2025 [1]. Digital transformation in civil aviation broadens the cyber threat landscape by exposing critical systems to significant vulnerabilities. Ten studies document that key aviation components, including communication, navigation, surveillance, and IT networks, lack robust security measures. For example, researchers report that wireless communication systems are inherently insecure [2] and that aeronautical communication standards rarely incorporate cybersecurity requirements [3]. Authors examining aircraft IT systems note substantial gaps in secure software design and communication practices [4], while studies targeting Air Traffic Management (ATM) propose extended threat models to capture interdependent risks [5]. An analysis of ten identified attack vectors indicates that seven bear high potential impact (e.g., GNSS spoofing, malware injection, ransomware, and Automatic Dependent Surveillance-Broadcast (ADS-B) exploitation) and that detection capabilities mostly remain limited or moderate. Complementary approaches, such as novel risk assessment frameworks [6] and threat taxonomies [7], illustrate efforts to systematically assess evolving challenges, including those emerging with urban air mobility and unmanned aerial systems [8].

This paper addresses three core Research Questions (RQ) to get insights into the threat landscape in civil aviation:

RQ1 *"Who are the relevant threat actors targeting civil aviation?"*

RQ2 *"What are the critical attack vectors exploited in this domain?"*

RQ3 *"How vulnerable are current aviation systems to these evolving threats?"*

A systematic review examines cybersecurity challenges across commercial aviation systems, encompassing both airborne and ground infrastructures [4], [9]. The analysis covers critical aviation components including Communication, Navigation, and Surveillance (CNS) systems, air-ground communication, radio navigation aids, and aeronautical surveillance systems [6]. The research methodology combines systematic reviews with both theoretical and empirical approaches [2], incorporating qualitative analysis through expert perspectives [10] and comprehensive threat assessments [5]. This multifaceted approach enables a thorough examination of cybersecurity vulnerabilities in modern aviation systems, from aircraft information technology to ATM infrastructure [4], [11].

The focus is on operationally plausible cyber risks where today's aviation stack lacks integrity and authenticity guarantees (e.g., unauthenticated navigation/communication links, long certification cycles limiting timely patching). The objective is a pilot-centred risk view aligned with International Civil Aviation Organization (ICAO) Doc 9859, complemented by a concise attacker capability model to bound feasibility claims.

The paper is structured as follows: Section I introduces the cybersecurity challenges in civil aviation, formulates the guiding research questions, and outlines the study's objectives. Section II reviews prior research on aviation cybersecurity, synthesizing existing findings on threat actors, attack vectors, and systemic vulnerabilities. Section III details the qualitative research methodology, combining a systematic literature review with semi-structured interviews of commercial airline pilots. In Section IV, we analyze the aviation threat landscape by characterizing relevant adversaries and their capabilities. Section V categorizes and evaluates twelve representative attack vectors across airborne, ground, and communication subsystems, mapping them onto ICAO Doc 9859 risk matrices. Section VI synthesizes pilot assessments with literature find-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

ings, highlights divergences in risk perception, and identifies critical technical, organizational, and regulatory gaps. Finally, Section VII concludes by addressing the research questions, summarizing the principal findings, and providing targeted recommendations alongside a roadmap for future research.

II. Related Work

The cybersecurity landscape in civil aviation has garnered increasing attention from researchers, with studies employing diverse methodological approaches to understand the evolving threat environment. This section synthesizes existing research on threat actors, attack vectors, and system vulnerabilities that inform our systematic assessment framework.

A. Methodological Approaches in Aviation Cybersecurity Research

Recent scholarship has employed varied research methodologies to examine aviation cybersecurity challenges. Systematic literature reviews have been the predominant approach, with ten studies utilizing this method to map the threat landscape [6], [7], [9], [11]–[15]. Complementing these reviews, nine studies have conducted empirical security analyses, including simulator-based assessments and experimental demonstrations [16]–[22]. Seven studies have incorporated threat modeling approaches to systematically analyze potential attack scenarios [7], [9], [14], [19]–[22].

B. Threat Actor Landscape

Previous studies consistently highlight that civil aviation is targeted by a heterogeneous set of adversaries with differing resources, motivations, and operational approaches. The most frequently cited categories include nation-state actors, cybercriminal groups, terrorists, insiders, and hacktivists [9], [11], [12]. These groups represent the spectrum of actors capable of exploiting aviation's technical and organizational vulnerabilities.

The literature portrays nation-states as the most capable adversaries due to their access to Advanced Persistent Threat (APT) capabilities, financial resources, and strategic objectives such as espionage or disruption of national infrastructure. Cybercriminal organizations, in contrast, primarily pursue financial gain through ransomware campaigns, data breaches, or extortion targeting airlines and airports. Hacktivist groups, although less technically advanced, are motivated by ideological agendas and often seek to disrupt operations or attract publicity through Denial of Service (DoS) attacks and website defacements [7], [23]. Insider threats, arising from employees, contractors, or vendors with privileged access, constitute a distinct category, as they bypass perimeter defenses and have intimate knowledge of internal systems [12], [24].

While these categories are well established in the literature, the relative emphasis differs across studies. Some works prioritize nation-state threats given their strategic reach [11], whereas others highlight the immediacy of ransomware and data theft by cybercriminals [9]. A detailed analysis of these threat actors, including their motivations, technical capabilities, and case illustrations, is provided in Section IV, where expert interviews are combined with the literature review to offer a more nuanced assessment.

C. Attack Vectors and Technical Methodologies

Existing research identifies a variety of attack vectors that exploit structural weaknesses across aviation's communication, navigation, and information systems. The most frequently discussed targets include wireless-based protocols, such as ADS-B and GNSS, traditional communication links, such as Aircraft Communications Addressing and Reporting System (ACARS) and Controller-Pilot Data Link Communications (CPDLC), and emerging digital integration platforms like System Wide Information Management (SWIM) [2], [7], [25]. While the specific technical mechanisms vary, the common denominator lies in the lack of authentication, encryption, or integrity controls across many of these protocols.

Methodologically, prior studies have employed simulation environments, red-teaming exercises, and protocol analysis to demonstrate the feasibility of such exploits [16], [21]. These works collectively show that low-cost tools (e.g., software-defined radios) are sufficient to interfere with air traffic surveillance, while malware injection and lateral movement remain feasible through poorly segmented IT infrastructures [19], [22]. Although most demonstrations have occurred under controlled research settings, the findings establish credible evidence that such attacks could be replicated in operational environments.

In summary, the literature confirms that attack vectors span airborne, ground, and communication subsystems, with varying degrees of technical sophistication and potential impact. Rather than duplicating these detailed findings here, Section V provides a structured risk categorization that integrates both expert interviews and literature insights. This approach ensures that the vectors most relevant to operational aviation security are systematically prioritized and evaluated.

To improve interpretability for readers unfamiliar with aviation-specific cyber-physical systems, selected attack vectors are briefly illustrated with simplified operational examples. For instance, spoofing-based attacks exploit the absence of authentication in radio-frequency navigation signals. In such scenarios, an adversary broadcasts forged signals with higher signal strength than legitimate sources, causing onboard systems to prioritize the malicious input. Similarly, message injection attacks against communication protocols (e.g., ADS-B or ACARS) involve crafting syntactically valid but false transmissions that are accepted by receiving systems due to missing integrity checks. These simplified examples demonstrate that many aviation attack vectors do not require physical system compromise but instead exploit trust assumptions in legacy protocol design.

D. System Vulnerabilities and Security Gaps

Aviation systems exhibit persistent vulnerabilities across multiple domains. Communication systems, including Very

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

High Frequency (VHF) radio, CPDLC, ACARS, and Satellite Communication (SATCOM), consistently lack "authentication, encryption, and integrity checks, making them susceptible to eavesdropping, spoofing, and message injection" [2], [9].

Navigation technologies present significant security gaps, with GNSS, ADS-B, ILS, and VHF Omnidirectional Range systems vulnerable to "jamming and spoofing due to unauthenticated signals" [2], [22]. ATM systems face risks from "integration of Information Technology and operational systems" that "increases exposure to cyber threats, with documented cases of system shutdowns and operational disruption" [9].

Avionics and onboard networks represent another vulnerability domain, with "interconnected systems, including Electronic Flight Bags, In-Flight Entertainment and Connectivity, and flight management systems" at risk from "both external and insider threats, especially when connected to public or less-secure networks" [7], [16].

III. Methodology

This research employs a qualitative methodology to investigate the cybersecurity threat landscape and vulnerabilities in civil aviation systems. The approach was chosen due to the complexity and sensitivity of the topic, which benefits from expert-based insights rather than purely quantitative data.

A. Research Design

The study follows an exploratory and descriptive design. The primary objective was to gather structured knowledge about realistic cyber threats, system vulnerabilities, and the expert perception of risk within the aviation domain. Given the limited publicly available data and operational sensitivity of aviation systems, expert interviews were deemed the most suitable method for data collection.

B. Expert Selection and Sampling

A purposive sampling strategy was employed to identify individuals with relevant professional expertise. Participants were recruited via professional networks of current airline pilots and training organizations. No incentives were offered. Prior to participation, all individuals received an information sheet and provided written informed consent, including permission for audio-recording and anonymized quotation. Selection criteria included:

- Professional background in civil aviation, ATM, or aircraft systems.
- Specific experience in cybersecurity, cyber risk assessment, or information security.
- Academic or consulting roles with publications or projects in the aviation cybersecurity field.

The experts represented organizations such as aviation authorities, cybersecurity consultancies, aviation software vendors, and research institutions. All participants had more than five years of relevant work experience.

C. Interview Design and Procedure

Semi-structured interviews were conducted to ensure comparability while allowing for flexible and in-depth discussion. The interview guide included thematic blocks related to:

- Identification of relevant threat actors.
- Perception of technical vulnerabilities in airborne and ground systems.
- Assessment of communication protocol security (e.g., ACARS, ADS-B).
- Evaluation of organizational cybersecurity challenges.

Interviews were conducted via video conferencing and lasted between 30 and 60 minutes. With consent, all sessions were recorded and transcribed.

The expert sample included also five aviation professionals serving as pilots/copilots on aircraft such as the Airbus A320, A330/340, and the CRJ-900. These individuals possessed an average flight experience ranging from 6,000 to 9,000 hours, indicating a high level of operational expertise. All participants were employed by German commercial airlines, providing a consistent organizational backdrop. The interviews focused on critical aspects of aviation cybersecurity, specifically addressing airborne systems, ground infrastructure, and communication links. GNSS, avionics systems, the ACARS, and threats related to Instrument Landing System (ILS) spoofing are examples.

D. Data Analysis

The transcribed material was analyzed using qualitative content analysis. An inductive coding scheme was developed to identify recurring themes. The analysis focused on clustering insights into threat types, attack feasibility, system weaknesses, and organizational practices. These categories informed the structure and content of the subsequent results and discussion sections.

While this study is grounded in qualitative analysis due to the domain's sensitivity and expert-driven nature, future extensions could explore quantitative methodologies such as the Common Vulnerability Scoring System (CVSS), probabilistic threat trees, or hybrid models in aviation. The assessed risks in this paper are based on preliminary efforts in this direction. Beyond thematic clustering, the interview material was also systematically translated into structured risk assessments using the ICAO Doc 9859 framework. Each pilot evaluated selected cyberattack scenarios, such as GNSS spoofing, avionics exploitation, and ILS spoofing, along two axes: probability (A = extremely unlikely to E = frequent) and severity (1 = negligible to 5 = catastrophic). Their judgments were mapped onto a 5×5 ICAO risk matrix, which is widely adopted in aviation safety management. This approach ensures comparability with established safety practices and enables triangulation between qualitative expert insights and quantitative-style classifications. The resulting matrices therefore capture how operational pilots perceive cyber risks in a manner directly aligned with industry safety protocols. The simplified versions (cf. Tables II-V) are related to pilot-centered studies using the mentioned ICAO

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Figure 1. Interlinked Systems in Civil Aviation [27]

Doc 9859-compliant matrices [26], offering a valuable foundation for quantitative risk propagation modeling in aviation cybersecurity.

E. Limitations of Sampling

While the pilot cohort provides deep operational expertise, the limited number of interviewees (n = 5) constrains statistical generalizability. The participants, however, were highly experienced professionals (6,000–9,000 flight hours on Airbus A320/330/340 and CRJ-900 aircraft in German commercial airlines), offering significant operational credibility. This yields high experiential validity but does not capture global variation in procedures, regulatory environments, or fleet compositions. Future studies should broaden the sample to include additional roles such as air traffic controllers, maintenance personnel, and regulatory authorities, as well as participants from different regions and airlines. Such expansion will strengthen external validity and provide a more comprehensive view of civil aviation cybersecurity.

IV. Threat Landscape

Understanding the diverse threat landscape is critical for developing effective cybersecurity strategies in civil aviation. Cybersecurity threats in aviation are evolving with the increasing digitization and integration of Information and Communication Technology (ICT) tools [11], smart technologies, and Internet of Things (IoT) devices in airports [5]. The aviation sector is confronted by a range of adversaries, each with different motives, capabilities, and targets. This section provides an overview of the primary threat actors and analyzes their motivations and technical capacities.

A. Overview of Threat Actors

Nation-state actors are considered among the most capable and persistent adversaries, pose significant risks to aviation systems, targeting communication networks and avionics for political influence and intelligence gathering [9], [28], [29]. Their motivations typically include political influence, economic disruption, and strategic intelligence gathering. These actors, often operating as APT groups, have access to substantial resources and engage in long-term operations [11]. The complexity and interconnectedness of CNS infrastructure amplify the potential scope of attacks [30]. This includes communication networks, air traffic control infrastructure, or avionics systems [11], [31].

Cybercriminals and hacktivists often exploit known vulnerabilities for financial gain or ideological purposes [5], [32]. Airports and airlines face various cyber risks, including potential loss of passenger information, disruption of operations, and damage to aircraft [33]. Ransomware attacks on airport IT systems and attempts to breach airline customer databases are common examples. Hacktivists may seek to disrupt flight operations or expose perceived injustices using defacement, DoS attacks, or information leaks [11], [34].

Insider threats pose a significant cybersecurity risk, often underestimated compared to external attacks [35]. These threats come from individuals with legitimate access, such as employees, contractors, or vendors, who may exploit their knowledge of internal systems and bypass perimeter defenses. The impact of insider attacks can be severe, as evidenced by high-profile cases at companies like Tesla and government agencies like the US Department of Defense [36]. Insiders are particularly dangerous due to their operational knowledge and ability to bypass conventional perimeter defenses [37].

B. Motivations and Capabilities

Threat actors targeting civil aviation operate with a broad spectrum of motivations:

- **Political motivations:** Aimed at destabilizing nations, projecting power, or coercing policy changes (nation-states) [38], [39].
- **Economic motivations:** Including theft of personal data, ransom payments, or illicit trade in sensitive information (cybercriminals) [5], [11].
- **Ideological motivations:** Related to activist agendas or grievances against the aviation industry (hacktivists) [31], [32].

Their capabilities vary widely. Nation-states can exploit zero-day vulnerabilities and conduct coordinated cyber-physical operations [40]. Cybercriminals often rely on off-the-shelf malware and social engineering, whereas insiders leverage their access privileges and domain familiarity to perform covert actions [37]. All actors are increasingly capable of targeting key aviation subsystems, including avionics, ground control centers, and satellite communication links [41].

Table I presents a summary of selected real-world aviation cybersecurity incidents that illustrate the feasibility and impact of documented vulnerabilities. These incidents illustrate how theoretical vulnerabilities manifest in operational environments. For example, the 2018 British Airways breach demonstrates the susceptibility of airline customer data infrastructures to large-scale data exfiltration, emphasizing risks associated with ground-based IT systems. Similarly, ransomware attacks on airport Maintenance and Repair Operations (MRO)

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

highlight how cyber incidents can disrupt operational continuity without directly affecting flight safety. More recent GNSS spoofing incidents in the Middle East provide empirical evidence of navigation manipulation in live environments, reinforcing concerns regarding unauthenticated positioning systems. Finally, attacks targeting airport IT infrastructure, such as the Hamburg airport incident, illustrate the increasing convergence of traditional IT threats with aviation-specific operational systems.

TABLE I
SELECTED CYBERSECURITY INCIDENTS IN CIVIL AVIATION

Year	Incident	Vector	Impact
2018	Brit. Airways breach	Customer DB	Data theft (£20M)
2020	Ransomware on ST Eng.	Airport MRO	Operations halt
2023	GNSS spoofing Iraq/Iran	GNSS	Position deviat.
2024	Hamburg airport cam. hack	IT system	Public data leak

V. ATTACK VECTORS AND VULNERABILITIES

This section synthesizes the findings from expert interviews with a literature-based analysis, providing a structured overview of the main attack vectors in civil aviation. It follows the categories of airborne systems, ground infrastructure, and communication links. The complex interplay between airborne, ground, and communication subsystems is illustrated in Figure 1, which highlights how interdependencies across aviation infrastructure expand the cyber attack surface.

The tables (cf. Tables II-IV) reflect the prioritization of threats as described by expert interviewees and supported by the literature review. In detail, the attack vectors discussed below are derived from a thematic synthesis of the expert interviews and the structured literature review. Key vulnerabilities were categorized where at least two interviewees identified similar risks, which were then cross-validated against peer-reviewed and industry literature. This integration follows principles of inductive qualitative coding and thematic saturation, ensuring methodological rigor in capturing domain knowledge. The likelihood and impact ratings are derived through a triangulated synthesis of literature and practitioner input. Attack vector tables are annotated with expert-based (E), literature-based (L), or combined evidence (E+L), to clarify the provenance of each rating.

All risk assessments presented in Tables II-IV and Figures 2-4 apply the ICAO Doc 9859 methodology (see Section III-D for details). This ensures comparability with established aviation safety practices.

A. Airborne Systems

Airborne systems include all onboard digital subsystems, such as avionics, navigation, and communication modules. Experts emphasized the high dependency of modern aircraft on complex, interconnected technologies, many of which were not originally designed with cybersecurity in mind. The summary of the risks for airborne systems is shown in Table II.

TABLE II
RISKS IN AIRBORNE SYSTEMS

Attack Vector	Likelihood	Impact	Evidence
Legacy avionics exploitation	Medium	High	E+L
SATCOM command injection	Medium	High	E+L
ADS-B ghost aircraft injection	High	High	E+L
GNSS spoofing/jamming	High	High	E+L

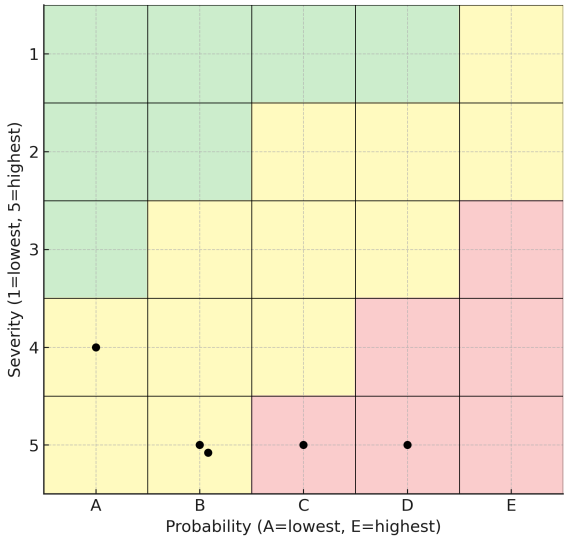


Figure 2. Pilot-based ICAO Doc 9859 Risk Classification of Avionics System Exploitation. Each dot represents one pilot's assessment (n = 5). The background coloring follows ICAO safety management conventions (green = low, yellow = medium, red = high).

1) Avionics and Flight Management Systems: Legacy avionics platforms often operate on proprietary or outdated software with limited patching capabilities due to certification constraints [42]. These systems are vulnerable to local and remote exploitation if attackers gain access to maintenance ports or use wireless vectors during pre-flight servicing [14], [43]. According to expert interviews and industry reports, exploitation is considered *moderately likely*, but the *impact is high* due to the proximity of these systems to flight-critical functions.

Pilot-based ICAO 9859 classifications support these findings. As shown in Figure 2, avionics exploitation was rated as a high-severity threat, with most pilots placing it in the medium- to high-risk category. This reflects the direct link between avionics compromise and loss of flight-critical functionality.

2) SATCOM and Data Links: SATCOM-based communication plays a central role in long-haul aviation [44]. Experts highlighted that many implementations lack strong encryption or robust authentication, exposing them to spoofing or hijacking attempts [45]. Attackers could theoretically disrupt the data integrity between cockpit and ground services or inject false control commands [44]–[46]. Although complex, such attacks are *technically feasible*, making the *likelihood medium* and the *impact high* due to the potential for operational disruption.

3) ADS-B and GNSS: The ADS-B protocol broadcasts aircraft position data in plaintext, without encryption or au-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

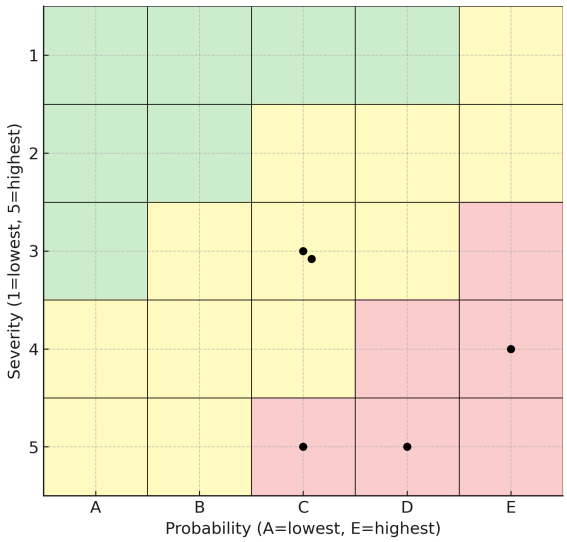


Figure 3. Pilot-based ICAO Doc 9859 Risk Classification of GNSS Spoofing Attacks. Dots indicate individual pilot evaluations (n = 5), clustering in the high-risk zone (red), reflecting the operational criticality of navigation spoofing.

thentication [45], [46]. This allows attackers to eavesdrop or inject ghost aircraft into air traffic visualizations [45], [47]. GNSS signals are also weak and susceptible to jamming or spoofing, which can mislead navigation systems [48], [49]. Several real-world incidents (Middle East) validate feasibility [31], [50], [51]. These attacks are *well-documented in research and red-teaming efforts*, making the *likelihood high* and the *impact high*.

From a technical perspective, GNSS spoofing attacks are typically executed using software-defined radio platforms that generate counterfeit satellite signals synchronized to legitimate GNSS transmissions. The attacker gradually increases signal power to override authentic signals ("carry-off" attack), thereby manipulating the receiver's computed position and velocity without triggering immediate alarms. Experimental studies have demonstrated that such attacks can be conducted with commercially available hardware and moderate expertise, highlighting the feasibility of real-world exploitation.

Pilot evaluations provide additional insights into the perceived risks of communication system exploitation. For example, Figure 3 shows the aggregated classification of GNSS spoofing attacks. All five respondents assigned medium-to-high probability and medium-to-high severity, placing this scenario consistently in the high-risk category. This confirms the criticality of GNSS spoofing already emphasized in research and real-world incident reports.

B. Ground Infrastructure

Ground-based systems provide essential support for aircraft operations, including logistics, passenger processing, and air traffic control. The following subsystems were identified as critical and the summary of the risks for ground infrastructure is shown in Table III.

TABLE III
RISKS IN GROUND INFRASTRUCTURE

Attack Vector	Likelihood	Impact	Evidence
Ransomware in airport IT	High	Medium	E+L
ATM network compromise	Medium	High	E+L
Third-party lateral movement	High	Medium	E
Manipulated maintenance records	Medium	High	E+L

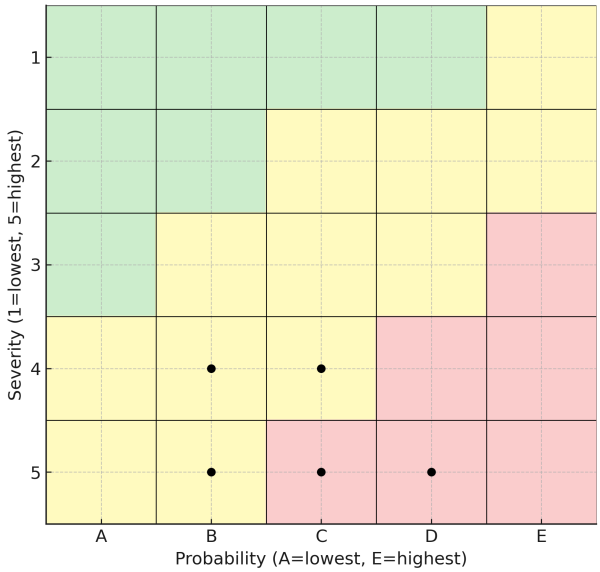


Figure 4. Pilot-based ICAO Doc 9859 Risk Classification of ILS Spoofing. Pilot assessments fall within medium-to-high risk classes (yellow/red), highlighting the potential safety and reputational consequences of manipulated landing guidance.

1) Airport IT Systems and Networks: Experts reported that airport IT systems are often heterogeneous and difficult to centrally manage [5]. Attack vectors include ransomware, spear-phishing, and lateral movement via third-party contractor access [31], [34]. Ransomware attacks have frequently occurred globally (e.g., ransomware at airport check-in systems) [52], [53], confirming the *high likelihood* also due to weak endpoint protection, though the *impact is considered medium* as the attacks typically affect business continuity rather than flight safety.

2) ATM Systems: Air traffic control environments rely on legacy architectures and software as well as centralized infrastructures [54]. Experts warned that insufficient network segmentation and outdated authentication mechanisms pose severe risks, especially when connected to supervisory control and data acquisition systems [55]. Experts cite complex vendor ecosystems [34]. Access is often poorly segmented, and attackers can escalate privileges across IT networks [56]. The *likelihood is medium* due to controlled access, while the *impact is high* because compromised ATM systems could disrupt national airspace by impacting flight routing, a safety-critical function. Operational disruption in ATM may lead to cascading scheduling effects that indirectly compromise flight safety and emergency response coordination.

In addition to the literature- and expert-based risk ratings, a

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

pilot-centered risk classification was conducted. Figure 4 illustrates the risk assessment for ILS Spoofing, highlighting that all pilots rated such attacks in the medium-to-high risk classes, primarily due to reputational and operational implications.

ILS manipulation attacks target the radio guidance signals used during aircraft approach and landing, specifically the localizer (lateral guidance) and glide slope (vertical guidance). An attacker can transmit counterfeit signals on the same frequency bands, creating false alignment information for the aircraft. While such attacks require higher technical precision and proximity to the airport environment compared to GNSS spoofing, research prototypes have demonstrated the feasibility of inducing misleading course deviations under controlled conditions. The absence of robust authentication mechanisms in legacy ILS infrastructure remains a key enabling factor.

3) Supply Chain and Maintenance Systems: Digital systems used for aircraft maintenance, such as electronic logbooks and maintenance management tools, were identified as vulnerable due to limited access control and shared interfaces [42], [56]. Digital systems are exposed via remote or wireless interfaces [57]. If exploited, could lead to incorrect repairs or overlooked issues that allow subtle manipulation of aircraft safety-related data [58]. The *likelihood is medium*, and the *impact is high* due to latent threats to airworthiness.

C. Communication Links

Cybersecurity vulnerabilities in communication links were a key concern across all interviews and the summary of the risks for communication links is shown in Table IV.

TABLE IV
RISKS IN COMMUNICATION LINKS

Attack Vector	Likelihood	Impact	Evi.
ACARS interception	High	Medium	E+L
ACARS message manipulat.	Medium	High	E+L
SWIM data injection	Medium	Medium–High	E+L
VHF/UHF spoofing or jam.	Low–Medium	Medium	E+L

1) ACARS and Voice Communications: The ACARS transmits sensitive data like flight plans, fuel status and weather updates over plaintext VHF or SATCOM channels [59], [60]. Experts warned of the ease with which such messages can be intercepted or crafted valid messages can be injected into systems using low-cost software-defined radios [45], [47]. This results in a *high likelihood* for data interception, and a *medium impact* as the intercepted data could support more targeted or disruptive attacks.

2) SWIM and IP-based Protocols: The increasing adoption of SWIM introduces standardized interfaces for data exchange between aviation actors using standardized APIs over Internet Protocol (IP) [54]. However, this integration also extends the attack surface, especially when IP-based protocols are used without end-to-end encryption [31], [61]. Improper authentication may allow false data exchange (e.g., flight status, weather) [45]. While no public exploitation is known to date, *expert opinion assessed the likelihood as medium* and the *impact as medium to high*, depending on the data affected.

D. Electronic Flight Bags

While not formally included in the structured interview set, one pilot identified attacks on Electronic Flight Bags (EFBs) as an emerging scenario. Incorrect performance calculations or manipulated digital charts could lead to unsafe take-off or approach configurations. This was classified as a medium-risk, with moderate impact on aircraft and operational safety. Although less severe than GNSS or avionics exploitation, this vector illustrates the vulnerability of consumer-grade IT devices increasingly integrated into cockpit operations.

E. Summary of Findings

The aggregated results from expert interviews and supporting literature indicate that cyber risks in civil aviation vary significantly across system categories in both likelihood and potential impact, cf. Table V.

Airborne systems, particularly those relying on outdated avionics or unauthenticated data broadcasts such as ADS-B, were consistently rated as having the highest impact, given their direct connection to flight safety and the limited ability to implement rapid updates due to certification constraints. However, due to more restricted physical and logical access, the likelihood of successful exploitation was generally considered medium to high. Pilot-based assessments confirmed this evaluation, with GNSS spoofing and avionics exploitation consistently placed in the medium-to-high risk categories of ICAO Doc 9859 matrices, underscoring their immediate operational relevance.

Ground infrastructure, encompassing airport IT, ATM networks, and maintenance systems, presented a higher likelihood of exploitation. This was attributed to the widespread use of Commercial Off-The-Shelf (COTS) components, heterogeneous networks, and extensive third-party integration. Although the immediate safety impact of attacks on ground systems may be lower, operational disruptions and indirect safety effects, such as delayed maintenance updates, elevate the risk severity. Pilot evaluations highlighted that ILS manipulation belongs to this category and classified it as medium-to-high risk, reflecting concerns over potential impacts on approach and landing safety.

Communication links were assessed to have a medium likelihood of exploitation due to known vulnerabilities in protocols like ACARS and the integration of IP-based systems like SWIM. Experts emphasized that while direct safety effects depend on the attack vector, compromised communication integrity could result in degraded situational awareness or operational delays. Pilot evaluations, however, tended to rank such scenarios lower in relative criticality, reflecting a stronger focus on threats with immediate flight safety implications rather than those primarily affecting efficiency or continuity.

VI. DISCUSSION

This study aimed to provide a comprehensive assessment of cybersecurity threats in civil aviation by combining expert insights with a structured vulnerability evaluation. Unlike prior studies, this paper introduces a layered risk prioritization based

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE V
RISK OVERVIEW BY ATTACK VECTOR CATEGORY

Category	Likelihood	Impact	Evi.
Airborne Systems	Medium–High	High	E+L
Ground Infrastructure	Medium–High	Medium–High	E+L
Communication Links	Low–High	Medium–High	E+L

on operational pilots' judgment integrated with literature-based scoring. The findings confirm that the aviation sector is facing a complex, multi-dimensional threat landscape characterized by both well-understood and emerging attack vectors.

A. Expert Consensus and Divergences

Across the interviews, there was broad consensus regarding the most critical vulnerabilities: unauthenticated communication protocols (especially ADS-B and ACARS), outdated avionics platforms, and ransomware threats in ground infrastructure, which is consistent with prior studies highlighting the lack of authentication and integrity protection in aviation communication systems [2], [9], [45]. Experts agreed that while the likelihood of attacking airborne systems may be lower due to restricted access and specialized knowledge requirements, the potential impact is significantly higher due to safety-critical dependencies. Conversely, ground systems are more exposed due to extensive third-party integration and reliance on legacy IT architectures, a trend also emphasized in airport cybersecurity and smart infrastructure studies [5], [53].

A distinguishing contribution of this study lies in its triangulated risk synthesis, which not only validates known attack vectors such as ADS-B spoofing and ACARS plaintext but also stratifies their risk severity based on direct expert evaluation. Unlike prior literature reviews, this paper systematically ranks vulnerabilities across airborne, ground, and communication layers, reflecting real-world operator prioritization and threat perception.

Notably, the perception of risk around communication systems varied more significantly among experts. Some emphasized the high exploitability of ACARS and voice channels due to lack of encryption, while others viewed such channels as low-priority targets, arguing that operational redundancy limits their criticality. This divergence points to the need for scenario-based risk modeling to clarify the consequences of link-level compromises.

The pilot risk classifications identified GNSS spoofing, avionics exploitation, and ILS manipulation as the most critical scenarios, consistently placing them in the medium-to-high risk range. These findings align with expert assessments of unauthenticated navigation and control systems but diverge regarding communication channels, which pilots generally perceived as less critical than ground experts did.

The application of ICAO-compliant risk matrices provides a methodological bridge between cybersecurity research and established aviation safety practices [6], [26]. Pilots emphasized threats that directly affect operational continuity and flight-critical functions, while ground-focused experts drew greater attention to organizational and IT-layer vulnerabilities with more indirect cockpit implications. Aligning pilot perspectives with ICAO Doc 9859 therefore illustrates how cyber threats can be systematically integrated into Safety Management Systems, offering regulators and operators a practical framework for embedding cyber risk awareness into aviation safety practice.

B. Identified Security Gaps

The results presented in Section IV highlight how cyber threats manifest across system layers. Rather than restating attack vectors, we group observed weaknesses into four overarching categories: (1) legacy systems lacking patchability, (2) insecure-by-design communication protocols, (3) unsegmented network architectures, and (4) limited visibility across Operational Technology (OT)/IT domains. These categories are not isolated; their interdependencies intensify systemic risk. For example, outdated avionics in airborne systems not only lack encryption but also interact with unverified ground data over ADS-B and SATCOM, compounding threat exposure.

- **Legacy technology**: Many components in avionics and air traffic systems remain unpatched or unsupported, yet are essential to certified aircraft and control operations.
- **Protocol weaknesses**: Unsecured communications (e.g., ADS-B, ACARS) are still in widespread use without industry-wide mandates for cryptographic protection [2], [45], [59].
- **Insufficient segmentation**: Airport and ATM networks often lack adequate isolation between OT and IT systems, allowing lateral movement in case of breach.
- **Limited situational awareness**: There is a notable gap in the deployment of real-time intrusion detection or anomaly recognition systems tailored for aviation environments.

Synthesizing the results reveals that many risks cannot be addressed at the subsystem level alone. Vulnerabilities in airborne platforms (e.g., legacy flight systems) are often mirrored by insecure communications (e.g., unencrypted ACARS) and exacerbated by permissive ground networks (e.g., shared maintenance IT). These layers form a tightly coupled threat surface where mitigation strategies must be holistic rather than component-specific.

C. Organizational and Regulatory Challenges

Beyond technical vulnerabilities, organizational barriers emerged as a dominant theme. Experts noted that aviation cybersecurity is hindered by inter-organizational complexity and unclear accountability across airline operators, airport authorities, Original Equipment Manufacturer (OEM), and regulators. This diffusion of responsibility contributes to delayed patch cycles, inconsistent incident reporting, and fragmented responses to shared threats.

From a regulatory perspective, initiatives such as the ICAO Cybersecurity Strategy and European Union Aviation Safety Agency (EASA) oversight programs have made progress in establishing a governance framework. However, practical

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

enforcement and harmonized adoption across countries and actors remain lacking. Several interviewees stressed the importance of moving from voluntary guidance to enforceable minimum cybersecurity baselines, which aligns with broader discussions in aviation cybersecurity governance literature [11], [31], [34].

D. Strategic Implications

Taken together, the results highlight the need for a layered and aviation-specific cybersecurity approach. Mitigation strategies must prioritize:

- Protection of safety-critical systems (e.g., navigation, control) through isolation and redundancy.
- Gradual deprecation of insecure communication protocols in favor of authenticated, encrypted alternatives.
- Continuous training and threat modeling across operational teams, IT personnel, and aircrew.
- Strengthening of information sharing platforms for threat intelligence between stakeholders.

The findings also support the adoption of advanced monitoring tools (e.g., AI-based intrusion detection) to detect anomalous patterns across OT and IT boundaries. While technical interventions are necessary, a cohesive security culture supported by policy and cross-organizational cooperation is essential to sustaining trust in civil aviation infrastructure.

VII. Conclusion and Future Work

A. Conclusion

The three research questions (RQ1-RQ3) were addressed by systematically identifying relevant threat actors, critical attack vectors, and the vulnerabilities of current systems. The answers are summarized below. In addressing RQ1, the study identified nation-states, cybercriminals, and insiders as the principal threat actors, consistent with previous aviation cybersecurity research [9], [11], [31]. Nation-state adversaries were considered the most capable, often motivated by political or strategic objectives, while cybercriminals targeted economic assets through extortion or data theft. Insider threats, although less visible, remain dangerous due to their system knowledge and access privileges.

For RQ2, the study systematically categorized attack vectors across three critical domains: airborne systems, ground infrastructure, and communication links. Particular emphasis was placed on unauthenticated data links such as ADS-B and ACARS, whose security weaknesses have been widely documented in prior studies [2], [45], [59]. These vectors were identified through both expert elicitation and evidence from past incidents, supporting their relevance and severity.

Addressing RQ3, the research showed that civil aviation systems remain highly vulnerable to cyberattacks, particularly due to legacy architectures, insufficient authentication mechanisms, and fragmented cybersecurity governance structures [6], [9], [11]. Vulnerabilities were not limited to outdated technologies, but also to organizational fragmentation and limited situational awareness across stakeholders. While the impact of an attack on airborne systems is typically higher due to safety implications, ground and communication systems were found to be more accessible, increasing the likelihood of compromise.

Importantly, the pilot-based assessments provide an operationally grounded complement to these findings. Pilots consistently rated GNSS spoofing, avionics exploitation, and ILS manipulation as high-risk scenarios within ICAO Doc 9859 risk matrices, aligning with expert and literature-based evaluations but emphasizing their immediate safety relevance. This perception highlights the gap between organizational assessments, which often prioritize IT- or infrastructure-based risks, and the cockpit perspective, which centers on flight-critical consequences.

By incorporating pilot judgments, the study demonstrates that effective cybersecurity in civil aviation requires not only technical and organizational countermeasures but also systematic integration of end-user expertise into risk classification. Such triangulation strengthens the validity of the findings and ensures that future mitigation strategies reflect both operational realities and broader systemic vulnerabilities.

B. Recommendations

Based on these results, the following recommendations are proposed. Each is explicitly linked to the corresponding research question to ensure coherence and traceability:

RQ1 (Threat Actors): Aviation regulators and industry actors should establish mandatory information-sharing frameworks and joint cyber exercises to enhance situational awareness and resilience across organizational boundaries. These measures address the complex threat actor landscape, ranging from state-sponsored APTs to insider threats, by strengthening collaborative defense mechanisms and reducing organizational silos. Pilot perspectives indicate that such initiatives should not only be cross-organizational but also extend to cockpit crews, who face the operational consequences of sophisticated adversaries. Training exercises that actively involve pilots in simulated cyber incidents could close the gap between strategic awareness and frontline response.

RQ2 (Attack Vectors): The adoption of secure communication protocols, such as encrypted ADS-B, IP-authenticated SWIM, and secure ACARS variants, must become mandatory. These steps directly mitigate attack vectors that exploit unauthenticated or plaintext messaging formats, which are prevalent in both airborne and ground-air communication systems. Pilot-based assessments emphasize the disproportionate risk posed by GNSS spoofing, avionics exploitation, and ILS manipulation. Therefore, mitigation strategies should prioritize these flight-critical domains, incorporating pilot feedback into the evaluation of new communication standards and ensuring that operational realities shape the prioritization of protocol upgrades.

RQ3 (System Vulnerability): Addressing systemic vulnerabilities requires both technical retrofitting and architectural modernization. Legacy avionics and airport IT systems must be upgraded using secure-by-design principles despite long certification cycles. In parallel, strict network segmentation

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

and anomaly detection tailored to OT/IT hybrid environments should be deployed to detect and contain threats across domain boundaries. These controls improve visibility into lateral movements and cross-layer attacks, especially in supply chain and maintenance subsystems. Pilot-based findings highlight that technological defenses should be complemented by enhanced training for aircrew, enabling them to recognize and respond effectively to anomalous navigation or communication behaviors that may indicate attacked vulnerabilities.

Strategic Roadmap for Mitigation: A phased roadmap is proposed to operationalize the recommendations and support implementation across the aviation sector. This roadmap spans short-, mid-, and long-term actions, corresponding to technical, infrastructural, and governance domains:

In the short term, priority should be given to securing vulnerable communication protocols. This includes deploying encryption and authentication mechanisms for ADS-B and ACARS transmissions, as well as enforcing strict access control policies for SWIM interfaces. Pilot input should be integrated into validation exercises to ensure these controls address operationally relevant attack scenarios.

In the mid term, focus must shift to infrastructure hardening. Key actions involve segmenting ATM and airport IT networks to reduce lateral movement risk, and upgrading airport systems that currently rely on outdated or unsupported software components. Pilot-centered risk classifications suggest that upgrades should be tested against flight-critical scenarios such as GNSS spoofing and ILS manipulation to verify their effectiveness in practice.

In the long term, sustainable cybersecurity in civil aviation requires robust governance mechanisms. These include mandatory coordinated vulnerability disclosure programs, harmonized reporting obligations across national aviation authorities, and alignment with international cybersecurity baselines as advocated by ICAO and EASA. Future governance models should institutionalize the integration of pilot perspectives into cybersecurity audits, certification cycles, and incident response frameworks to ensure long-term resilience.

C. Future Work

Future research should proceed along a phased roadmap connecting technical innovation with regulatory adoption.

a) Short-term (1–2 years): Focus on protocol hardening. Priority actions include deploying encryption and authentication for ADS-B and ACARS transmissions and enforcing strict access controls for SWIM interfaces. Pilot input should be integrated into test campaigns to ensure operational relevance.

b) Mid-term (3–5 years): Prioritize infrastructure hardening and modernization. Network segmentation within ATM environments and updates to outdated airport IT systems are essential. These upgrades should be validated against high-risk scenarios such as GNSS spoofing and ILS manipulation to confirm safety-critical effectiveness.

c) Long-term (5+ years): Establish robust governance mechanisms. This includes mandatory coordinated vulnerability disclosure, harmonized reporting obligations across ICAO and EASA, and institutionalized integration of pilot perspectives into certification cycles and incident response frameworks.

Parallel research should advance on three trajectories: (i) probabilistic risk models and AI-based anomaly detection for GNSS and avionics; (ii) AI-driven intrusion detection tailored to safety-critical avionics and ATM systems, accounting for certification constraints; and (iii) comparative governance studies across aviation, maritime, and rail domains to identify transferable best practices. Collectively, this roadmap strengthens both technical resilience and regulatory pathways, ensuring sustained cybersecurity in civil aviation.

REFERENCES

[1] A. Lawall, "Cybersecurity in Civil Aviation – Threat Landscape and Vulnerability Assessment of Attack Vectors," in *The Second International Conference on IoT-AI (IoTAI) 2025*, 2025, pp. 16–22. [Online]. Available: https://www.thinkmind.org/library/IoTAI/IoTAI_2025/iotai_2025_1_40_20038.html

[2] M. Strohmeier, M. Schäfer, R. Pinheiro, V. Lenders, and I. Martinovic, "On perception and reality in wireless air traffic communication security," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1338–1357, 2016.

[3] N. Mäurer, T. Guggemos, T. Ewert, T. Gräupl, C. Schmitt, and S. Grundner-Culemann, "Security in digital aeronautical communications a comprehensive gap analysis," *International Journal of Critical Infrastructure Protection*, vol. 38, p. 100549, 2022.

[4] M. Wolf, M. Minzlaff, and M. Moser, "Information technology security threats to modern e-enabled aircraft: A cautionary note," *Journal of Aerospace Information Systems*, vol. 11, no. 7, pp. 447–457, 2014.

[5] G. Lykou, A. Anagnostopoulou, and D. Gritzalis, "Smart airport cybersecurity: Threat mitigation and cyber resilience controls," *Sensors*, vol. 19, no. 1, p. 19, 2018.

[6] A. A. Elmarady and K. Rahouma, "Studying cybersecurity in civil aviation, including developing and applying aviation cybersecurity risk assessment," *IEEE Access*, vol. 9, pp. 143 997–144 016, 2021.

[7] E. Habler, R. Bitton, and A. Shabtai, "Assessing aircraft security: A comprehensive survey and methodology for evaluation," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–40, 2023.

[8] A. C. Tang, "A review on cybersecurity vulnerabilities for urban air mobility," in *AIAA Scitech 2021 forum*, 2021, p. 0773.

[9] G. Dave, G. Choudhary, V. Sihag, I. You, and K.-K. R. Choo, "Cyber security challenges in aviation communication, navigation, and surveillance," *Computers & Security*, vol. 112, p. 102516, 2022.

[10] C. A. P. Viveros, "Analysis of the cyber attacks against ads-b perspective of aviation experts," *Master's thesis, University of Tartu*, 2016.

[11] E. Ukwandu, M. A. Ben-Farah, H. Hindy, M. Bures, R. Atkinson, C. Tachtatzis, I. Andonovic, and X. Bellekens, "Cyber-security challenges in aviation industry: A review of current and future trends," p. 146, 2022.

[12] L. Kelly, "Threats to Civilian Aviation since 1975," 2021.

[13] L. Florido-Benítez, "The types of hackers and cyberattacks in the aviation industry," *Journal of Transportation Security*, vol. 17, no. 1, p. 13, 2024.

[14] E. Habler, R. Bitton, and A. Shabtai, "Evaluating the security of aircraft systems," *arXiv preprint arXiv:2209.04028*, 2022.

[15] L. Florido-Benítez, "Identifying and classifying cyberattacks on airports," *Cyber Security: A Peer-Reviewed Journal*, vol. 8, no. 1, pp. 63–79, 2024.

[16] M. Smith, M. Strohmeier, J. Harman, V. Lenders, and I. Martinovic, "Safety vs. security: Attacking avionic systems with humans in the loop," *arXiv preprint arXiv:1905.08039*, 2019.

[17] S. M. Zeigler, A. C. Hou, J. Martini, D. M. Norton, B. Phillips, M. Schwille, A. Strong, and N. Vest, "Acquisition and use of manpads against commercial aviation: Risks, proliferation, mitigation, and cost of an attack," Tech. Rep., 2019.

[18] F. Abozaid and A. Baydoun, "Resilience of aviation cyber-physical systems to malicious attacks," in *2024 International Conference on Computer and Applications (ICCA)*. IEEE, 2024, pp. 1–9.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[19] A.-V. Predescu and T. H. Stelkens-Kobsch, "Aviation security lab: A testbed for security testing of current and future aviation technologies," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*. IEEE, 2022, pp. 1–5.

[20] J.-P. Buch, R. Geister, L. Canzian, G. Gamba, and O. Pozzobon, "What the Hack Happened to the Flight Deck: Analyzing the Impact of Cyber Attacks on Commercial Flight Crews," in *AIAA Scitech 2019 Forum*, 2019, p. 0060.

[21] P. M. Berges, B. A. Shivakumar, T. Graziano, R. Gerdes, and Z. B. Celik, "On the feasibility of exploiting traffic collision avoidance system vulnerabilities," in *2020 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2020, pp. 1–6.

[22] R. M. Geister, J.-P. Buch, D. Niedermeier, G. Gamba, L. Canzian, and O. Pozzobon, "Impact study on cyber threats to GNSS and FMS systems," 2018.

[23] F. Francy, "The aviation information sharing and analysis center (a-isac)," in *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)*. IEEE, 2015, pp. 1–14.

[24] M. Strohmeier, M. Schäfer, M. Smith, V. Lenders, and I. Martinovic, "Assessing the impact of aviation security on cyber power," in *2016 8th International Conference on Cyber Conflict (CyCon)*. IEEE, 2016, pp. 223–241.

[25] S. L. Burton, "Mitigating advanced social engineering threats: multi-factor authentication and the evolving tactics of scattered spider in the aviation sector," *Brazilian Journal of Business*, vol. 7, no. 3, p. e81057, Jul. 2025. [Online]. Available: https://ojs.brazilianjournals.com.br/ojs/index.php/BJB/article/view/81057

[26] I. Doc, "ICAO Doc 9859 safety management manual," *Montreal: International Civil Aviation Organization*, 2018.

[27] P. Gontar, H. Homans, M. Rostalski, J. Behrend, F. Dehais, and K. Bengler, "Are pilots prepared for a cyber-attack? a human factors approach to the experimental evaluation of pilots' behavior," *Journal of Air Transport Management*, vol. 69, pp. 26–37, 2018.

[28] A. Pawlicka, M. Choraś, and M. Pawlicki, "Cyberspace threats: not only hackers and criminals. raising the awareness of selected unusual cyberspace actors-cybersecurity researchers' perspective," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020, pp. 1–11.

[29] D. Craigen, N. Diakun-Thibault, and R. Purse, "Defining cybersecurity," *Technology Innovation Management Review*, vol. 4, no. 10, 2014.

[30] Y. Xie, A. Gardi, and R. Sabatini, "Cybersecurity risks and threats in avionics and autonomous systems," in *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, 2023, pp. 0814–0819.

[31] A. Malatras, Z. Stanic, I. Lella, R. De Sousa Figueiredo, E. Tsekmezoglou, M. Theocharidou, R. Naydenov, and A. Drougkas, "Enisa threat landscape: Transport sector (january 2021 to october 2022)," 2023.

[32] J. Soldatos, J. Philpot, and G. Giunta, "Cyber-physical threat intelligence for critical infrastructures security: a guide to integrated cyber-physical protection of modern critical infrastructures," 2020.

[33] A. Alsaidi, A. Gutub, and T. Alkhodaidi, "Cybercrime on Transportation Airline," *Journal of Forensic Research*, vol. 10, pp. 1–4, 11 2019.

[34] B. I. Scott, "Aviation Cybersecurity: Regulatory Approach in the European Union," 2019.

[35] J. Predd, S. L. Pfleeger, J. Hunker, and C. Bulford, "Insiders behaving badly," *IEEE Security & Privacy*, vol. 6, no. 4, pp. 66–70, 2008.

[36] G. Mazzarolo and A. D. Jurcut, "Insider threats in cyber security: The enemy within the gates," *arXiv preprint arXiv:1911.09575*, 2019.

[37] B. Bean, "Mitigating Insider Threats in the Domestic Aviation System: Policy Options for TSA," 2017.

[38] R. Abeyratne, "Cyber Terrorism and Aviation - National and International Responses," pp. 337–349, 2011.

[39] A. F. Brantly, *A Fierce Domain: Conflict in Cyberspace, 1986 to 2012 edited by Jason Healy: Arlington, VA: Cyber Conflict Studies Association*. Taylor & Francis, 2014.

[40] M. C. Libicki, *Conquest in cyberspace: national security and information warfare*. Cambridge University Press, 2007.

[41] K. Sampigethaya and R. Poovendran, "Aviation cyber-physical systems: Foundations for future aircraft and air transport," *Proceedings of the IEEE*, vol. 101, no. 8, pp. 1834–1855, 2013.

[42] R. De Cerchio and C. Riley, "Aircraft Systems Cyber Security," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*. IEEE, 2011, pp. 1C3–1–1C3–7. [Online]. Available: https://doi.org/10.1109/DASC.2011.6095969

[43] L. Bogoda, J. Mo, and C. Bil, "A systems engineering approach to appraise cybersecurity risks of cns/atm and avionics systems," in *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, 2019, pp. 1–15.

[44] S. Khare and N. S. Talwandi, "Satellite communication vulnerabilities and threat landscape," in *2024 IEEE Silchar Subsection Conference (SILCON 2024)*. IEEE, 2024, pp. 1–6.

[45] M. Strohmeier, V. Lenders, and I. Martinovic, "On the security of the automatic dependent surveillance-broadcast protocol," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1066–1087, 2014.

[46] M. Schäfer, V. Lenders, and I. Martinovic, "Experimental analysis of attacks on next generation air traffic communication," in *Applied Cryptography and Network Security: 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings 11*. Springer, 2013, pp. 253–271.

[47] A. Costin, A. Francillon *et al.*, "Ghost in the air (traffic): On insecurity of ads-b protocol and practical attacks on ads-b devices," pp. 1–12, 2012.

[48] T. Humphreys, "Statement on the vulnerability of civil unmanned aerial vehicles and other systems to civil gps spoofing," *University of Texas at Austin (July 18, 2012)*, pp. 1–16, 2012.

[49] M. L. Psiaki and T. E. Humphreys, "Gnss spoofing and detection," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.

[50] D. Sullivan, "Above us only stars: Exposing gps spoofing in russia and syria," https://c4ads.org/publications/above-us-only-stars, 2019, c4ADS Report [retrieved: February, 2025].

[51] GPSJam.org, "Global gps interference map - crowdsourced gnss disruption data," https://gpsjam.org, 2024, [Retrieved: April, 2025].

[52] A. Lawall and P. Beenken, "A threat-led approach to mitigating ransomware attacks: Insights from a comprehensive analysis of the ransomware ecosystem," in *Proceedings of the 2024 European Interdisciplinary Cybersecurity Conference*, ser. EICC '24, S. Li, K. Coopamootoo, and M. Sirivianos, Eds. New York, NY, USA: Association for Computing Machinery, 2024, p. 210–216. [Online]. Available: https://doi.org/10.1145/3655693.3661321

[53] N. Koroniotis, N. Moustafa, F. Schiliro, P. Gauravaram, and H. Janicke, "A holistic review of cybersecurity and reliability perspectives in smart airports," *IEEE Access*, vol. 8, pp. 209 802–209 834, 2020.

[54] R. Medina, "Air Traffic Management (ATM)," in *Annales des Mines-Réalités industrielles*, no. 2. Cairn/Cairn, 2024, pp. 22–27. [Online]. Available: https://doi.org/10.3917/rindu1.242.0022

[55] K. Stouffer, S. Lightman, V. Pillitteri, M. Abrams, and A. Hahn, "Guide to Industrial Control Systems (ICS) Security," National Institute of Standards and Technology, Tech. Rep. 82, 2011.

[56] I. Ghafir, J. Saleem, M. Hammoudeh, H. Faour, V. Prenosil, S. Jaf, S. Jabbar, and T. Baker, "Security threats to critical infrastructure: the human factor," *The Journal of Supercomputing*, vol. 74, pp. 4986–5002, 2018.

[57] A. Mosenia and N. K. Jha, "A comprehensive study of security of internet-of-things," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 586–602, 2016.

[58] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[59] X. Lu, "Research on the security of communication addressing and reporting system of civil aircraft," in *IOP Conference Series: Earth and Environmental Science*, vol. 295, no. 3. IOP Publishing, 2019, p. 032026.

[60] A. Roy, "Secure aircraft communications addressing and reporting system (acars)," in *20th DASC. 20th Digital Avionics Systems Conference (Cat. No. 01CH37219)*, vol. 2. IEEE, 2001, pp. 7A2–1.

[61] C. Finke, J. Butts, R. Mills, and M. Grimaila, "Enhancing the security of aircraft surveillance in the next generation air traffic control system," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 1, pp. 3–11, 2013.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Professional Experience May Matter More Than LLM Support when Selecting and Prioritizing Cybersecurity Requirements

Damjan Fujs, Damjan Vavpotič, Tomaž Hovelja and Marko Poženel
University of Ljubljana
Faculty of Computer and Information Science
Ljubljana, Slovenia
e-mail: {damjan.fujs | damjan.vavpotic| | tomaz.hovelja| marko.pozenel}@fri.uni-lj.si

***Abstract*—This paper explores how access to Large Language Models (LLMs) and varying levels of professional software development experience affect the prioritization of cybersecurity requirements for web applications. Twenty-three postgraduate students (research participants) participated in a research study to prioritize security requirements (SRs) using the MoSCoW method and subsequently rated their proposed solutions against multiple evaluation criteria. We divided participants into two groups (one with and the other without access to LLM support during the task). Results showed no significant differences related to LLM use, suggesting that access to LLMs did not noticeably influence how participants evaluated cybersecurity solutions. However, statistically significant differences emerged between experience groups for certain criteria, such as estimated cost to develop a feature ($p < 0.05$), perceived impact on user experience ($p < 0.05$), and risk assessment related to non-implementation of the proposed feature ($p < 0.05$ and $p < 0.01$). Postgraduate students with more professional experience tended to provide higher ratings for user experience impact and lower risk estimates.**

Keywords-security requirements engineering; experiment; prioritization; estimation.

I. Introduction

This paper is an extended version of our conference paper entitled *Selecting Cybersecurity Requirements: Effects of LLM Use and Professional Software Development Experience* published at the Tenth International Conference on Cyber-Technologies and Cyber-Systems (CYBER 2025), September 28–October 2, 2025, Lisbon, Portugal [1].

Software development is inherently dynamic [2], pushing organizations to adopt or tailor development methodologies to remain efficient and competitive [3]. Prioritization of cybersecurity requirements, especially when assisted by Large Language Models (LLMs) or shaped by prior experience, takes place within this evolving context, where structured yet adaptable decision-making is essential. Our study, therefore, addresses this crucial area. As systems grow increasingly complex and interconnected (as well as powered with Artificial Intelligence (AI) [4]), cybersecurity has become a critical concern that must be addressed early in the development lifecycle [5]. Most decisions, we can argue, are probably still made by people. In practice, it is generally accepted that longer professional experience contributes to more effective decision-making. For example, Carvalho and Conte [6] report that software architecture decision making is predominantly driven by practitioners' professional experience. However, Franke and Buschle [7] explicitly test whether professional experience influences decision-making and find no significant association. This example of mixed evidence motivates us to get a closer examination of how professional experience shapes software engineering decisions in requirement engineering tasks. Moreover, given the mixed evidence on the role of professional experience and the uncertain benefits of current tool support, it is natural to ask whether alternative paradigms for decision support could help.

Accordingly, some emerging ideas [8]–[10], suggest that agentic AI systems could take on decision-making roles in specific areas of cybersecurity to address evolving cyber threats. This perspective somehow aligns with broader discussions on the integration of advanced AI systems (i.e., LLMs) into critical domains. For example, recent work in healthcare or medicine [11] highlights that leveraging LLMs promises substantial advancements but simultaneously introduces technical, ethical, and regulatory challenges, particularly regarding data privacy, deployment choices, and organizational dependence on external vendors. The main goal of this work is not to discuss the pros and cons of using agent systems. Instead, it focuses on the empirical test of "how" students use LLMs to create and prioritize security requirements.

At this stage of the study, we focus on whether there are statistically significant differences in how selected Security Requirements (from here on referred to as SR/SRs as plural) are perceived by participants who used an LLM versus those who did not. Specifically, we were interested in how participants estimated selected SRs across various evaluation criteria. This raises a broader and timely question: Can LLMs (or generative AI more broadly) begin to narrow or even erase the gap typically attributed to experience? The purpose of this paper is not to definitively answer the question posed above, but rather contribute empirical insights and provide guidelines for future empirical research in the field of software engineering or software development. In addition, we test the proposed hypotheses using postgraduate students as participants. There are two main reasons for that. First, students represent the next generation of software development professionals and constitute a valid population for empirical software engineering research. Prior work shows that studies involving students are an effective and widely accepted means to advance software engineering theories and technologies [12], particularly in controlled settings focusing

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

on cognitive and decision-making tasks. Second, students are in general typically familiar with using LLMs. For example, Boguslawsky, Deer, and Dawson [13] indicate that students are already highly familiar with LLMs and routinely integrate them into software development tasks.

Based on all the above, we hypothesize:

- **H1**: Access to a LLM has a significant effect on how participants rate their proposed SRs across the given evaluation criteria.
- **H2**: Professional experience with software development has a significant effect on how participants rate their proposed SRs across the given evaluation criteria.

Based on the proposed hypotheses, this paper offers two key contributions:

- **C1**: Empirical insight into the limited impact of LLMs on cybersecurity decision-making among postgraduate students. The study provides evidence that access to LLM support does not significantly influence how individuals prioritize or evaluate SRs.
- **C2**: Demonstration of the role of professional software development experience in prioritizing and evaluating SRs among postgraduate students. The study shows that professional software development experience significantly affects how postgraduate students assess cost, user experience, and risk, highlighting the importance of practitioner expertise in shaping effective cybersecurity strategies.

The rest of the paper is organized as follows. Following this Introduction and Background section, Section II briefly highlights existing related works. Section III highlights the research methodology used. In Section IV, we present the results and discuss them briefly. The conclusion and future works are presented in Section V. Finally, in Section VI we point out the limitations of our study.

II. Related Work

This section summarizes prior research related to our study. It provides background on existing approaches and findings, and helps place our work within the broader research landscape.

The creation of software requirements is a fundamental activity in any software engineering project and is traditionally recognized as a labor-intensive, human-driven process [14]. Recent advances in AI, particularly the development of LLMs, have introduced new possibilities for supporting software engineering tasks such as SR engineering. Such tools for task automatization have the potential to assist software engineers in activities such as eliciting, refining, and documenting requirements, potentially reducing manual effort and cognitive load. Umar and Lano [15] observe that automated support for RE has the potential to shorten software development lifecycles and reduce associated costs. However, their actual impact on higher level decision making tasks, such as prioritizing and evaluating SRs, remains insufficiently understood.

Prior research has explored various factors influencing the prioritization and evaluation of software and cybersecurity requirements, including tool support and individual expertise. Ronanki et al. [16] investigated the potential of ChatGPT to assist requirements elicitation. They found that requirements generated using ChatGPT were of higher quality than those generated by human requirements engineering experts. A similar observation was provided by Krishna et al. [14] where they found that LLMs can produce output comparable in quality to that of an entry-level software engineer when generating a software requirements specification. While general software requirements engineering has been extensively studied, particularly in terms of specification quality, tool support, and the role of human expertise, SR represents a specialized subset that introduces additional complexity. For instance, in the study of Perry et al. [17], they found that participants who had access to an AI assistant wrote significantly less secure code than those without such support, raising concerns about overconfidence in automated tools in security-critical tasks. Zhu et al. [18] provide a literature review of the application of LLMs in software security, focusing primarily on tasks such as bug detection, bug triage, and unit test generation. In addition, they systematically characterize prior studies by analyzing how LLMs were used, including the prompting strategies employed (e.g., zero-shot, few-shot), the target programming languages (e.g., JavaScript, Python, Java), and the underlying base models (e.g., GPT-2, GPT-3 Codex). Similarly, Cheng et al. [19] survey existing research on the application of generative AI techniques in requirements engineering. In their literature review, they find that LLMs are most frequently applied to requirements elicitation and analysis, and least commonly used for requirements management and validation.

Moreover, previous study from Jalali, Siegel and Madnick [20] did not find precise evidence that professional experience significantly shapes decision making in cybersecurity. In general, defining professional experience in software development is complex, as it encompasses diverse roles and learning paths, and it is similar in the field of cybersecurity. Baltes and Diehl [21] have shown that developers' self-assessments of expertise are highly context-dependent. Vadlamani and Baysal [22] suggest, that while both knowledge and experience are necessary components of software development expertise, they are not sufficient on their own, as soft skills are also important. Moreover, recent work of Jelen et al. [23] shows that students generally have limited real-world requirements engineering experience prior to graduation and often view it less favorably, whereas alumni emphasize its importance in practice. These findings underscore that experience is unevenly developed at the undergraduate level and context-dependent in how it shapes judgment.

The above mentioned studies highlight the role of AI tools and developer expertise in software engineering, yet little is known about how these factors influence the prioritization and evaluation of SRs. This study addresses the gap by examining the combined effects of LLM access and professional experience on cybersecurity decision-making among postgraduate

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

students.

III. Research Methodology

This section describes how the study was designed and conducted. We explain the study setting, participant selection, task design, data collection, and analysis approach.

We employed a controlled experiment [24] in our research conducted in May 2025. The participants in the experiment were postgraduate students taking a course in Advanced software development methodologies, which, is offered at the University of Ljubljana, Faculty of Computer and Information Science. The course is attended by students from technical disciplines who are enrolled in various master's programs, including Computer Science and Mathematics, Computer Science, and Multimedia. In the academic year 2024/2025, 48 students were enrolled in the course, and 23 chose to participate in the research, yielding a response rate of 47.9%.

Figure 1 represents the entire research framework that consists of three main phases (e.g., Survey, Task and Analysis). The first phase involves conducting a survey. The second phase is an experiment, in which participants complete a predefined task using a structured template. The final phase focuses on data analysis, including statistical testing to assess the significance between different groups and the reporting of median values.

The research was carried out in May 2025 and took approximately two class hours (around two hours) to complete, including instructions and all assigned tasks. The study was conducted on-site in a supervised classroom setting. This arrangement allowed participants to ask clarification questions during the task and enabled the researchers to monitor task execution, including adherence to the LLM usage condition, thereby ensuring consistency and internal validity of the experimental procedure.

A. Survey

In the first phase (*Survey*), we received informed consent from the participants in the study, explained the course of the research to them, and gave them instructions. As part of the survey, in the first phase, we collected basic data about their studies and professional experience with software engineering. The exact question for years of professional experience with software engineering was: "Excluding education, how many years have you been 'professionally' involved in software development (e.g., student work, project work, etc.)?".

B. Task

In the second phase (*Task*), respondents were assigned to groups. Namely, 23 research participants were divided into two groups; one group could use any LLM for the task (experimental group, N = 12), while the other could not (control group, N = 11). Both groups had the typical time available for practicals (i.e., 2 hours, including our instructions). Participants were assigned to the experimental and control groups randomly upon entering the classroom.

We prepared a scenario and a structured template for participants to enter their decisions into (See Figures 2 and 3). The scenario was that, as part of their work on the project (as part of the course, they were developing software to support ScrumBan [25]), they were tasked with identifying 15 SRs appropriate for enhancing the system's overall security posture. We limited participants to 15 SRs in order to establish a unified framework while reflecting the resource constraints commonly encountered in industry settings. While developing the ScrumBan web application, students gained some experience with security aspects, particularly through implementing the login user story. The implementation of the login user story required them to handle authentication mechanisms, such as enforcing password policies (e.g., minimum length of 8 characters, inclusion of various character types and numbers). Additionally, they could improve the login procedure by implementing optional enhancements, such as a password strength meter or similar features, which further encourage consideration of usability and security. In short, students were given a defined degree of autonomy in how features were implemented. While the course instructors assumed the role of product owners and guided the overall direction of the work, the technical implementation was largely left to the students. In this phase, students were free to choose their own implementation approaches, such as selecting appropriate libraries or adapting existing solutions to fit their design decisions and project constraints. The choice of development frameworks and programming languages was left entirely to the students as well.

The 15 SRs, initially identified by the participants, were subsequently prioritized using the MoSCoW method [26]. The objective was to select $2X$ 'Must-have', $2X$ 'Should-have', and $2X$ 'Could-have' features from the set of 15. The remaining nine mechanisms were categorized as 'Won't have this time'. A similar prioritization approach was used in Fujs et al. [27]. The primary rationale for this approach was to simulate resource constraints, thereby encouraging students to make prudent decisions when selecting security mechanisms. This reflects real-world practice, where security measures must often be prioritised through a form of triage. In addition, established security frameworks explicitly recognise that it is neither feasible nor desirable to implement all available controls; instead, they advocate selecting and tailoring a subset of mechanisms based on context and risk (e.g., OWASP - Open Worldwide Application Security Project ASVS - Application Security Verification Standard [28], NIST - National Institute of Standards and Technology Risk Management Framework [29]). Figure 2 shows an example of the completed prioritisation template, illustrating how postgraduate students proposed 15 potential security features and selected six of them as Must-have, Should-have, and Could-have requirements.

The final step of the second task involved evaluating the six prioritized features using predefined criteria, as shown in Table I. Figure 3 shows an example of a completed estimation template, in which postgraduate students assessed the six previously prioritised security features by providing estimates

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

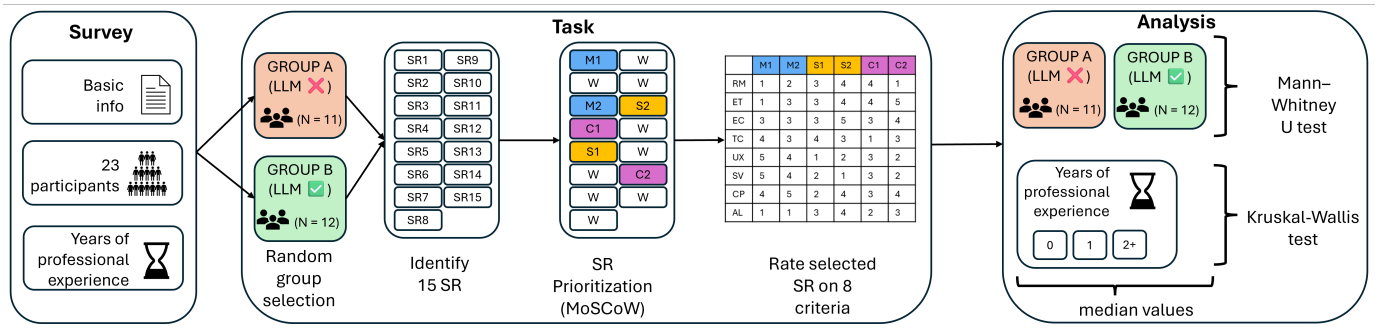


Figure 1. The research framework consists of three main phases (e.g., Survey, Task and Analysis). Notes: LLM (Large Language Model), SR (Security Requirements).

Table 1. Notes: MoSCoW (M=Must-have, S=Should have, C=Could-have, W=Won't have this time. The gray text (#0) represents an example. You can use ONLY 2X Must have, ONLY 2X Should-have, ONLY 2X Could-have. The rest is Won't have this time. This is because resources are typically limited in organizations.

ID	Functionality / Security feature	MoSCoW	Short argumentation/justification of priorities
#0	Password strength meter	Could-have	Improves user awareness but doesn't directly enforce security. Useful, not essential.
#1	Database security	Must-have	The database must be securely protected to prevent unauthorized access. Only authorized users with proper credentials should be able to perform insert, update, or delete operations.
#2	Data loss recovery	Must-have	No backup or recovery plan for your Scrum data.
#3	New profile creation	Should-have	When an admin profile is created, the user should not be able to see a list of all existing accounts. Additionally, the system should support the creation and management of organizations.
#4	File upload security	Should-have	Allowing file uploads (attachments, avatars) without proper validation, which might lead to malware upload
#5	Error Handling in Asynchronous Functions	Could-have	All asynchronous function calls should include proper error handling to catch and manage exceptions gracefully.
#6	Role Checking by User ID	Could-have	Role verification should be performed based on the user's unique ID rather than their username for improved security and accuracy.
#7	Audit logging	W	Detailed audit trails of user actions (who did what, when, and from where).
#8	User privacy controls	W	Allow users to control visibility of their projects, tasks, and profiles.
#9	Security Awareness & Training	W	Provide guidelines for users on creating secure passwords and recognizing phishing
#10	Behavioral Analytics & Anomaly Detection	W	Use machine learning or heuristic algorithms to detect unusual user behavior.
#11	Fine-Grained Permission Scopes	W	Beyond simple role-based access, support permission scopes (e.g., read-only vs. read-write vs. admin on specific projects or sprints).
#12	Immutable Task and Sprint History	W	Allow rollback to previous versions or snapshots.
#13	Content Moderation & Filtering	W	Implement moderation tools for user-generated content (comments, attachments).
#14	Rate Limiting & Brute Force Protection	W	Apply rate limits on login attempts, API calls, and critical actions.
#15	Security Incident Response Automation	W	Automate initial containment steps (lock accounts, revoke tokens) upon detecting breaches.

Figure 2. Example of a completed template, in which postgraduate students identified 15 potential security features and prioritised six of them using the MoSCoW method (2 Must-have, 2 Should-have, and 2 Could-have). Note: W denotes "Won't have this time"; row "#0" is instructional only.

across multiple criteria (risk if missing, time, cost, technical complexity, UX impact, security value, production criticality, and abuse likelihood) using a standardised five-point scale.

As part of this step, we aimed at gathering additional quantitative data regarding the rationale behind the participants' prioritization decisions. We used a 5-point scale ranging from 1 to 5 for each evaluation criterion. For example, in the case of Estimated Time (ET), participants were asked to assess how long it would take to implement an overdue feature, where option/value 1 corresponded to "less than 1 hour" and option/value 5 to "more than 10 hours." Intermediate options (i.e., 2, 3, and 4) were intentionally omitted to avoid over-constraining their responses and to encourage clearer distinctions in judgment. The study was conducted on-site at the university, allowing us to control whether participants were placed in a group with access to an LLM for the task or not. Additionally, we ensured that participants could ask questions if any part of the instructions was unclear.

Figures 4 and 5 show the instructions that the students received immediately before the experiment.

C. Analysis

In the last phase (*Analysis*), we analyzed the collected data. We used appropriate non-parametric statistical tests [30] given the sample size of 23 respondents. Specifically, the Kruskal-Wallis test was used to examine whether there were significant differences between the medians of three or more independent groups, while the Mann-Whitey U test was used to assess significantdifferences between the medians of two groups [31]. Specifically, we employed the Mann-Whitney U test to assess whether there were statistically significant differences in prioritizations based on whether respondents did or did not use LLMs. Furthermore, respondents who were allowed to use LLMs had complete freedom to choose the

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Table 2: Standardized scale (1-5) for all metrics. See instructions for explanation of scales for each category. The gray boxes represent an example of what is filled in for #0 (password meter above - a fictional example). Here, don't enter "Won't have this time" (W)! You only need to have 6 lines/rows (as provided in the table above).

Security Features	Risk if missing	Estimated Time	Estimated Cost	Technical complexity	UX impact	Security Value	Critical for production	Abuse Likelihood
#0	2	1	1	1	5	2	1	1
#1 Must-have	5	4	3	4	1	5	5	5
#2 Must-have	5	3	2	3	1	4	5	2
#1 Should-have	3	2	1	2	3	3	2	3
#2 Should-have	4	3	2	3	2	4	3	4
#1 Could-have	2	2	1	2	3	2	2	1
#2 Could-have	3	2	1	3	1	3	2	3

Figure 3. Example of a completed estimation template showing how postgraduate students evaluated the six selected security features across a set of predefined metrics using a five-point scale. The criterion is presented on Table I.

TABLE I. THE CRITERION WITH EIGHT ITEMS, BY WHICH RESPONDENTS EVALUATED THEIR SELECTED PRIORITY SECURITY MECHANISMS.

ID	*Item*	*1 - Lowest*	*5 - Highest*
1 RM	Risk if not implemented	Minimal risk if not implemented	Critical security risk if not implemented
2 ET	Estimated time	Less than 1 hour	More than 10 hours
3 EC	Estimated cost	No cost, trivial to implement	High cost, external tools or experts needed
4 TC	Technical complexity	Very simple, can be done without research	Very complex, requires redesign or specialized knowledge
5 UX	UX impact	Almost no user impact on UX	High impact on user UX
6 SV	Security value	Adds minimal security benefit	Essential for application security
7 CP	Critical for production	Not needed for launch	Absolutely necessary before production release
8 AL	Abuse likelihood	Very unlikely to be abused	Very likely to be abused without this feature

LLM of their choice. Most chose the version of ChatGPT available at the time (N = 6), followed by DeepSeek (N = 2), Gemini (N = 2), Perplexity (N = 1), and Claude (N = 1). To examine differences across varying durations of professional experience, we used the Kruskal-Wallis test [30], suitable for comparing two or more groups. Based on these non-parametric tests, we then reported the Median. Based on their experience with professional software development, participants were divided into three groups: the first group included participants with zero years of experience (N = 10), the second group included participants with one year (N = 6), and the third group included participants with two or more years of experience (N = 7). This grouping was based on a qualitative judgment, as the participants were postgraduate students who were not yet formally employed. However, some had gained relevant professional software development experience through internships, freelance work, or other informal roles.

IV. RESULTS AND DISCUSSION

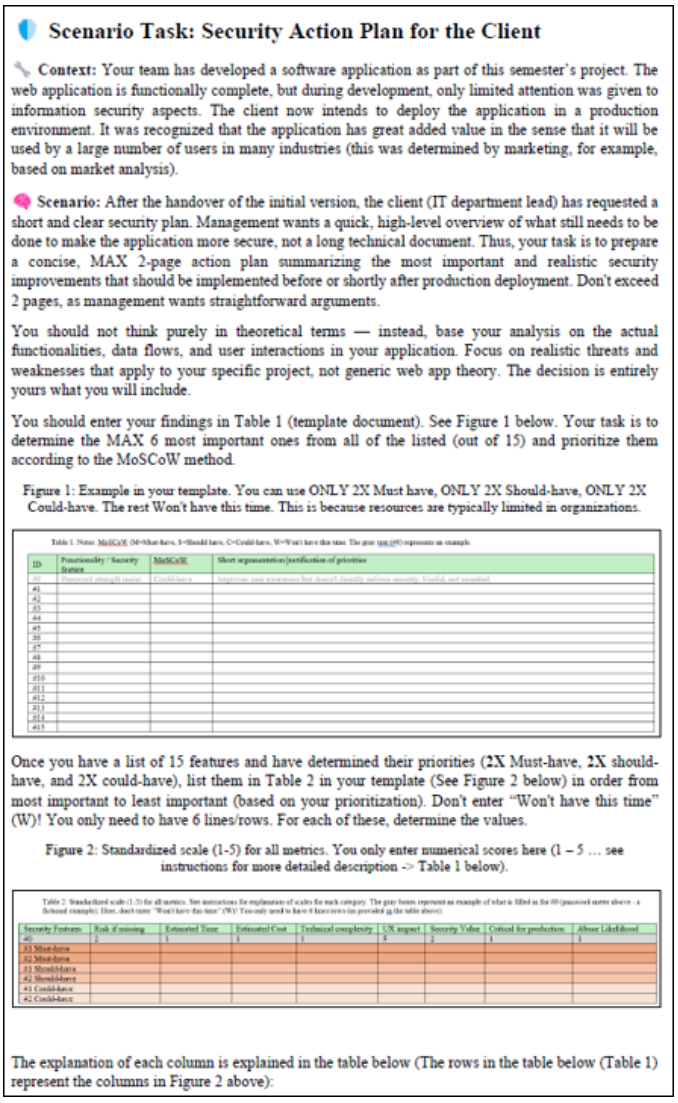
In this section we present the results of the data analysis and discuss them in relation to the proposed hypotheses. We first report the statistical outcomes for each hypothesis and then interpret what these results indicate. The discussion focuses on explaining the observed patterns, comparing them with related work, and clarifying what can and cannot be concluded based on the data.

Respondents selected up to six SRs using the MoSCoW prioritization method and subsequently rated each feature based on eight predefined criteria (see Figure 3). This resulted in a total of 48 ratings per respondent (8 criteria × 6 prioritized SRs). Figure 1 illustrates this rating step within the overall process and provides an example of the rating form used to assess the selected SRs across the eight criteria ("Rate selected SR on eight criteria").

The Mann-Whitney U test revealed no statistically significant differences across any of the evaluation criteria (column *item* in Figure 1). Based on these results, we conclude that access to an LLM did not significantly influence how respondents rated their proposed SRs. Therefore, Hypothesis H1 is not supported. Because we did not find significant differences, we do not report descriptive statistics (e.g., medians) for these comparisons. A possible explanation for the lack of statistically significant differences is that the LLM primarily served as a support tool for generating SRs, rather than influencing how participants evaluated their own solutions. Since the ratings were based on self-assessment, they were likely shaped more by the respondents' individual understanding, confidence, or prior knowledge than by the presence or absence of the LLM. Furthermore, given that the participants were postgraduate students with limited formal industry experience, many may have lacked the expertise to critically evaluate the quality of their proposed SRs. As a result, their assessments may have been similar across groups, regardless of LLM access.

Pavlič et al. [32] studied user story effort estimation in agile environments, comparing development teams that had assistance in generative AI tools to control teams without such support (i.e., conventional effort estimation). Contrary to our findings, they found statistically significant differences between regular and AI-assisted teams. However, it is also worth noting that in our case, it is not the same problem domain, as our respondents evaluated their own SRs (based on eight criteria), while the study participants in Pavlič et al. [32] evaluated the effort in pre-prepared user stories. Moreover, it is important to take into account the fact that in our case, the use of LLM was an option for the experimental group (i.e., we did not force the experimental group to necessarily use LLM). We intended to create a setting that approximates real-world

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org



Scenario Task: Security Action Plan for the Client

Context: Your team has developed a software application as part of this semester's project. The web application is functionally complete, but during development, only limited attention was given to information security aspects. The client now intends to deploy the application in a production environment. It was recognized that the application has great added value in the sense that it will be used by a large number of users in many industries (this was determined by marketing, for example, based on market analysis).

Scenario: After the handover of the initial version, the client (IT department lead) has requested a short and clear security plan. Management wants a quick, high-level overview of what still needs to be done to make the application more secure, not a long technical document. Thus, your task is to prepare a concise, MAX 2-page action plan summarizing the most important and realistic security improvements that should be implemented before or shortly after production deployment. Don't exceed 2 pages, as management wants straightforward arguments.

You should not think purely in theoretical terms — instead, base your analysis on the actual functionalities, data flows, and user interactions in your application. Focus on realistic threats and weaknesses that apply to your specific project, not generic web app theory. The decision is entirely yours what you will include.

You should enter your findings in Table 1 (template document). See Figure 1 below. Your task is to determine the MAX 6 most important ones from all of the listed (out of 15) and prioritize them according to the MoSCoW method.

Figure 1: Example in your template. You can use ONLY 2X Must have, ONLY 2X Should-have, ONLY 2X Could-have. The rest Won't have this time. This is because resources are typically limited in organizations.

Once you have a list of 15 features and have determined their priorities (**2X** Must-have, **2X** should-have, and 2X could-have), list them in Table 2 in your template (See Figure 2 below) in order from most important to least important (based on your prioritization). Don't enter "Won't have this time" (W)! You only need to have 6 lines/rows. For each of these, determine the values.

Figure 2: Standardized scale (1-5) for all metrics. You only enter numerical scores here (1 – 5 … see instructions for more detailed description -> Table 1 below).

The explanation of each column is explained in the table below (The rows in the table below (Table 1) represent the columns in Figure 2 above):

Figure 4. The first page of the instructions students received for the assignment.

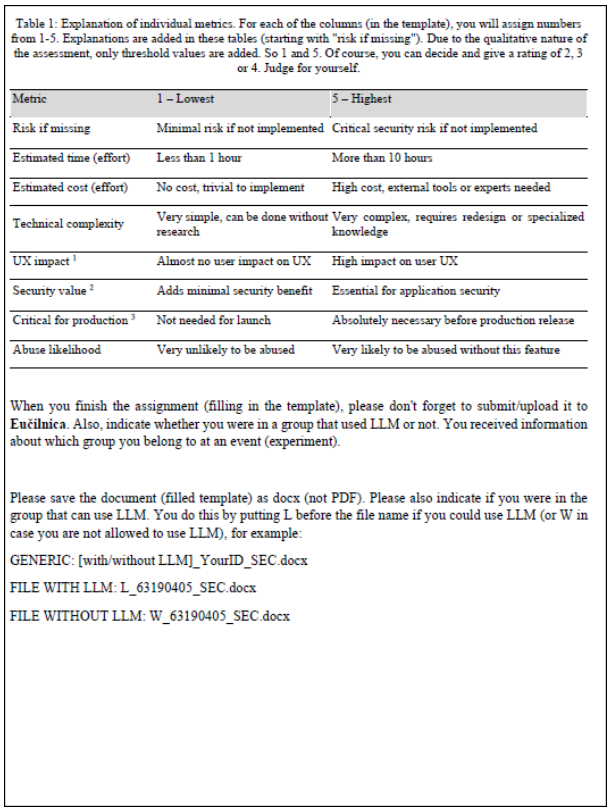


Table 1: Explanation of individual metrics. For each of the columns (in the template), you will assign numbers from 1-5. Explanations are added in these tables (starting with "risk if missing"). Due to the qualitative nature of the assessment, only threshold values are added. So 1 and 5. Of course, you can decide and give a rating of 2, 3 or 4. Judge for yourself.

Metric	1 – Lowest	5 – Highest
Risk if missing	Minimal risk if not implemented	Critical security risk if not implemented
Estimated time (effort)	Less than 1 hour	More than 10 hours
Estimated cost (effort)	No cost, trivial to implement	High cost, external tools or experts needed
Technical complexity	Very simple, can be done without research	Very complex, requires redesign or specialized knowledge
UX impact [1]	Almost no user impact on UX	High impact on user UX
Security value [2]	Adds minimal security benefit	Essential for application security
Critical for production [3]	Not needed for launch	Absolutely necessary before production release
Abuse likelihood	Very unlikely to be abused	Very likely to be abused without this feature

When you finish the assignment (filling in the template), please don't forget to submit/upload it to **Učilnica**. Also, indicate whether you were in a group that used LLM or not. You received information about which group you belong to at an event (experiment).

Please save the document (filled template) as docx (not PDF). Please also indicate if you were in the group that can use LLM. You do this by putting L before the file name if you could use LLM (or W in case you are not allowed to use LLM), for example:

GENERIC: [with/without LLM]_YourID_SEC.docx

FILE WITH LLM: L_63190405_SEC.docx

FILE WITHOUT LLM: W_63190405_SEC.docx

Figure 5. The second page of the instructions students received for the assignment.

industry conditions, where access to a given technology, such as an LLM, is available. Still, its actual use remains at the discretion of the individual.

To test hypothesis H2, we conducted a Kruskal-Wallis test [30], which revealed statistically significant differences for specific evaluation items. Table II shows five items where statistically significant differences in scores occurred for certain prioritized SRs. Items, for which no statistically significant differences have been found are not shown in Table II (there were 43 such items). This result neither conclusively supports nor definitively refutes the hypothesis, as statistically significant differences were found for some items but not for most. However, it suggests that professional experience in software development may have an influence on certain evaluation criteria.

The results indicate that statistically significant differences were found in the prioritization of should-have and could-have SRs, while no such differences were observed for must-have SRs. One possible explanation is that must-have SRs represent fundamental security mechanisms that are universally expected in any system (in addition, we also presented various cybersecurity mechanisms within the course, such as the OWASP (Open Worldwide Application Security Project) ASVS - Application Security Verification Standard [33]). Furthermore, students were provided with a predefined user story for authentication, which already required the implementation of basic login security features (e.g., secure password handling, user option to change the passwords, etc.). Additionally, participants may have based their decisions on the specific

TABLE II. MEDIAN VALUES FOR ITEMS BY YEARS OF PROFESSIONAL EXPERIENCE WITH SOFTWARE DEVELOPMENT. P-VALUES INDICATE STATISTICAL SIGNIFICANCE FOR THE ITEM (ID).

ID	*Years of professional experience*	*Median*	*p-value*
$S1EC$	none (0)	2.00	0.010
	1 year	2.50	
	2+ years	3.00	
$S1UX$	none (0)	1.00	0.036
	1 year	1.00	
	2+ years	2.00	
$S2RM$	none (0)	4.00	0.049
	1 year	4.00	
	2+ years	3.00	
$C1RM$	none (0)	3.00	0.003
	1 year	2.00	
	2+ years	3.00	
$C2EC$	none (0)	1.50	0.018
	1 year	2.50	
	2+ years	2.00	

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

characteristics of the web application they developed, leading to more consistent prioritization in this category.

C1RM achieved a p-value < 0.01, while S1EC, S1UX, S2RM and C2EC achieved a p-value < 0.05. In addition, it can also be observed that out of the eight criteria, statistically significant differences occur in three types, namely: Estimated Cost (EC), UX Impact (UX), and risk if not implemented (RM). Note that we were not interested in what actual SRs the respondents proposed, but rather in their values - that is, their assessments according to the criteria (see Table I). Among these criteria, estimated cost (EC) stands out most prominently in both S1 and C2. The results show that participants without professional experience significantly underestimated the anticipated cost of developing a proposed feature. This could be due to limited exposure to real-world development constraints such as budgeting, resource allocation, or integration complexity. They may focus mainly on what the security feature should do, rather than on the additional work required to build and maintain it. In contrast, more experienced participants likely drew from hands-on experience in estimating effort and understanding hidden development costs. It is also important to note that estimated costs were assessed using a relative scale rather than absolute monetary values. As a result, participants may compare SRs in terms of perceived effort or complexity, rather than actual budget figures. This may have further encouraged less experienced students to rely on intuition rather than concrete cost considerations, while more experienced students were better able to map the scale to real-world development effort.

In S1UX, participants with two or more years of professional experience stand out by assigning a higher median rating to the impact of the proposed feature on user experience. Similarly, in S2RM, participants with two or more years of professional experience provided slightly lower median estimates of the risk associated with not implementing the proposed feature, compared to those with no experience or only one year of experience.

An important observation from the study (i.e., from students' solutions) is that the majority of proposed security mechanisms were predominantly technical in nature. Participants primarily focused on mechanisms such as authentication, encryption, access control, logging, or other system-level safeguards, while comparatively fewer solutions addressed organizational, procedural, or human-centered aspects of information security.

For example, among the 345 identified SRs (23 students × 15 SRs), only a small portion (1.7%) can be classified as non-technical security mechanisms. The list below presents selected examples of security features proposed by the students. Features prefixed with "LLM" indicate that they were identified by participants who had access to an LLM during the task. For each security feature, the justification is reproduced verbatim from the students' own explanations of why they considered the SR relevant, while the priority reflects the MoSCoW classification assigned by the respective respondent.

- **LLM Security feature 1: "Frequent password changes"**. *Justification*: "In case their password gets leaked. But is also typically hated by users". *Priority*: Won't have this time.
- **LLM Security Feature 2: "User education pop-ups (phishing warnings)"**. *Justification*: "While valuable for security awareness, these can wait until after core security features are implemented. *Priority*: Won't have this time.
- **Security feature 3: "Pick an ISO security standard and conform to it"**. *Justification*: "Standards can be good guides for added security." *Priority*: Won't have this time.
- **Security feature 4: "Add Terms of Service"**. *Justification*: "Ask the user to comply to our terms of service, so we have basis to ban them and not get sued." *Priority*: Won't have this time.
- **Security feature 5: "Security awareness training"**. *Justification*: "Regularly educate employees about cybersecurity risks and best practices so that they can implement a robust system." *Priority*: Won't have this time.
- **Security feature 6: "Informing employees about secure coding practices"**. *Justification*: "Well-educated and informed programmers write better and more secure applications." *Priority*: Won't have this time.

Notably, all six were assigned the lowest priority ("Won't have this time"), indicating that non-technical measures were consistently ranked lower in priority in favor of technical solutions. This pattern suggests that participants predominantly conceptualized cybersecurity as a technical problem to be addressed through system-level controls, while organizational, procedural, or human-centered measures were viewed as secondary or optional. These efforts point to a bigger shift in thinking, and means that cybersecurity should be built into information systems from the start of software development lifecycle, not added later with security patches. Still, creating so-called security by design is not just a technical task, it also depends on people, processes, and the organization as a whole [34].

This finding is consistent with existing cybersecurity literature, which notes a preference toward technical security mechanisms in both research and practice. Security mechanisms are often conceptualized and implemented as technical countermeasures [35] embedded within software or infrastructure, whereas non-technical measures, such as user training, organizational policies, or other socio-technical interventions, tend to receive less attention. Moreover, prior work has also highlighted that this technical-centric perspective can lead to an incomplete treatment of cybersecurity as a whole [27][36], as many real-world security incidents stem from human [37], organizational [38], or process-related factors rather than purely technical flaws.

If we speculate, the tendency of students to prioritize technical solutions may also be explained by the educational and experiential background of the participants. As postgraduate students with limited industry experience, respondents are likely to be more familiar with technical controls explicitly covered in computer science and software engineering curric-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

ula. In contrast, organizational or socio-technical security measures often require exposure to real-world contexts, such as enterprise risk management, security operations, or regulatory compliance, which students may not yet have encountered. and that, as a result, participants may have perceived technical mechanisms as more concrete, actionable, or "correct" solutions, and therefore defaulted to them when asked to propose security requirements. Another reason may be that technical security mechanisms are perhaps easier to articulate, verify, and justify within an academic or design-oriented setting. Unlike organizational or behavioral measures, technical mechanisms can be directly mapped to system functionality and are commonly documented in security standards, guidelines, and best-practice frameworks. This may have reinforced participants' inclination to frame SRs primarily as technical design problems rather than as socio-technical ones.

Moreover, Among the proposed non-technical SRs, only two are strictly related to non-technical aspects, namely Security Features 6 and 7, as they explicitly refer to training. The remaining security features are primarily technical in nature, although some have indirect non-technical implications. For instance, password changes are fundamentally a technical measure but may also involve user behavior.

A practical implication is that, when making important decisions during the requirements engineering phase, additional experts should be involved, as they can provide alternative perspectives on the requirements. A similar approach is common in medicine, where the development of specialized software requires contributions from different expert profiles (e.g., epidemiologists, computer scientists, IT specialists, physicians, etc.) [39]. The same applies to the development of web applications where engaging experts with cybersecurity expertise is advisable.

We use the modal verb "may" in the title to reflect appropriate scientific caution. Although our results indicate that professional experience shows a clearer association with SRs prioritization than LLM support, these findings are bound by the study context, sample size, and participant population. The wording, therefore, acknowledges uncertainty and context-dependence while accurately summarizing the directional evidence observed in this study.

V. Conclusion and Future Works

In our research, 23 postgraduate students took part in a study aimed at prioritizing SRs using the MoSCoW method. Afterward, they evaluated their proposed solutions against several criteria. The postgraduate students were split into two groups: one group had access to LLM support during the task, while the other did not.

In our study, we found that access to LLM did not significantly influence how participants prioritized SRs. However, professional software development experience played a notable role in shaping evaluations. Participants with more experience rated the impact on user experience higher and perceived lower risks associated with not implementing certain features. Significant differences were also observed with estimated cost, user experience, and risk assessment, highlighting the importance of domain expertise in cybersecurity decision-making.

While the current study provides valuable insights into the use of LLMs for evaluation tasks, several opportunities remain for further exploration. Future research should consider designing tasks that require deeper interaction with the model to better evaluate its potential impact for "evaluation tasks".

There are several avenues that future research or researchers can pursue. First, future studies could strengthen the evaluation by incorporating external expert assessments or structured peer reviews to obtain more objective judgments of solution quality.

Second, future work could move beyond aggregate ratings and examine the content and quality of the proposed solutions in greater detail, such as which SRs were identified, how comprehensively they address relevant vulnerabilities, and how effectively they were prioritized. Analyzing the specific mechanisms proposed, the threat coverage they provide, and their alignment with established security best practices would allow for a more fine-grained comparison of solution quality.

Third, future research could also investigate how different professional roles interact with and evaluate model-generated outputs. For example, developers may primarily assess technical correctness and implementation feasibility, project managers may focus on delivery timelines and resource constraints, and other stakeholders may emphasize strategic value, risk reduction, and return on investment. Examining these role-specific perspectives could provide deeper insight into how such outputs are interpreted and applied in practice, and could inform evaluation approaches that better reflect real-world, multi-stakeholder decision-making.

Fourth, another promising direction is to explore the impact of contextual information on evaluation outcomes. For instance, providing participants with explicit organizational constraints, threat models, or regulatory requirements may influence how SRs are identified and prioritized, leading to more realistic or context-aware assessments.

Finally, future studies could consider cross-domain or cross-industry comparisons. This means evaluating whether participants from different sectors (e.g., finance, healthcare, public administration, etc.) prioritize solutions differently could shed light on domain-specific expectations, risk perceptions, and cybersecurity practices.

VI. Limitations

While the results provide useful insights, they should be interpreted with care. In this section, we outline several limitations of the study that may influence the findings. We also briefly explain why certain methodological choices were made, often due to practical constraints or to better reflect real-world settings. These limitations and clarifications are included to support a transparent interpretation of the results and to clarify the scope, within which the conclusions can be drawn.

While we can see some differences based on our data analysis, it is difficult to argue about the influence of profes-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

sional software development experience and the use of LLM based on these results alone. Thus, some limitations should be considered in the interpretation of these findings.

First, the number of respondents is relatively small, limiting the findings' statistical power and generalizability. With fewer participants, individual responses can have a stronger impact on the overall findings, and smaller differences between groups may not become visible. However, this is largely due to the fact that the study was conducted within a single course, which has a limited number of enrolled students. As such, the sample size reflects the practical constraints of the study setting and is typical for course-based or exploratory research. The results should therefore be interpreted as indicative rather than definitive.

Second, although certain trends emerge, for instance, more experienced participants assigning higher user experience impact or lower risk estimates, these differences may also reflect individual interpretation or subjective biases rather than consistent effects of professional experience.

Third, the ratings are self-reported, and participants may have relied on intuition or heuristics rather than systematic analysis, further complicating the interpretation. Therefore, while the data suggest a potential link between experience and how participants assess different aspects of cybersecurity features, these observations should be interpreted with caution. However, self-reports are common in early studies and can still reveal useful patterns, especially when the goal is to capture first impressions and decision habits.

Fourth, a potential selection bias may have occurred during group selection, as participants were assigned based on their position within the computer classroom (we counted and placed the first 11 individuals present in one group and the remaining 12 in another). This method may have unintentionally clustered individuals with similar characteristics, such as higher academic achievement, thereby affecting group comparability. However, this grouping selection approach was chosen for practical reasons and helped ensure a transparent and reproducible assignment process. It allowed participants to be allocated quickly and consistently, without introducing subjective decisions by the researchers.

Fifth, another limitation concerns the nature and depth of LLM integration. Participants may not have fully utilized the LLM's capabilities due to time constraints, unfamiliarity with prompting, or skepticism about the tool's relevance, etc. However, this was a deliberate design choice. Participants were given full freedom in how and whether they used the LLM, in order to reflect realistic practice settings where such tools are optional and not tightly prescribed.

Acknowledgments

This work was funded by the Slovenian Research and Innovation Agency [Grant Number P2–0426]. The authors thank the participants in the research for their eager collaboration. We would also like to thank the Editor for handling our paper and the reviewers for their insightful comments, which helped to improve the paper.

Declarations

Assist. Prof. Dr. Damjan Fujs is a member of the Editorial Board of International Journal On Advances in Security. He had no involvement in the editorial handling of this manuscript or the peer-review process.

References

[1] D. Fujs, D. Vavpotič, T. Hovelja, and M. Poženel, "Selecting cybersecurity requirements: Effects of LLM use and professional software development experience," IARIA CYBER 2025, The Tenth International Conference on Cyber-Technologies and Cyber-Systems, Sep. 2025, pp. 13–17. DOI: 10.48550/arXiv.2510.04274.

[2] C.-Y. Chen, P.-Y. Hsu, and H.-N. Vu, "Collaborative process tailoring in evolutionary software development: A teamwork-quality perspective," *Software Quality Journal*, vol. 31, no. 1, pp. 89–119, 2023. DOI: 10.1007/978-3-030-70873-3_36.

[3] A. Mihelič, S. Vrhovec, B. Markelj, and T. Hovelja, "Delegation-based agile secure software development approach for small and medium-sized businesses," *IEEE Access*, pp. 189 611–189 635, 2024. DOI: 10.1109/ACCESS.2024.3514889.

[4] E. Letier and A. Van Lamsweerde, "Obstacle analysis in requirements engineering: Retrospective and emerging challenges," *IEEE Transactions on Software Engineering*, pp. 795–801, 2025. DOI: 10.1109/TSE.2025.3534318.

[5] T. Jungebloud, N. H. Nguyen, D. D. Kim, and A. Zimmermann, "Model-based structural and behavioural cybersecurity risk assessment in system designs," *Computers & Security*, p. 104 543, 2025. DOI: 10.1016/j.cose.2025.104543.

[6] L. Carvalho and T. Conte, "Software architecture decision-making process: The practitioners' view from the Brazilian industry," *Science of Computer Programming*, vol. 244, p. 103 302, 2025. DOI: 10.1016/j.scico.2025.103302.

[7] U. Franke and M. Buschle, "Experimental evidence on decision-making in availability service level agreements," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 58–70, 2015. DOI: 10.1109/TNSM.2015.2510080.

[8] N. Kshetri, "Transforming cybersecurity with agentic AI to combat emerging cyber threats," *Telecommunications Policy*, p. 102 976, 2025. DOI: 10.1016/j.telpol.2025.102976.

[9] T. Raheem and G. Hossain, "Agentic AI systems: Opportunities, challenges, and trustworthiness," in *2025 IEEE International Conference on Electro Information Technology (eIT)*, IEEE, 2025, pp. 618–624. DOI: 10.1109/eIT64391.2025.11103638.

[10] V. Obojo, R. A. Ikuesan, and E. Iortom, "A critical evaluation of agentic AI in cybersecurity: The pros and the cons," in *2025 9th Cyber Security in Networking Conference (CSNet)*, IEEE, 2025, pp. 1–5. DOI: 10.1109/CSNet67572.2025.11288238.

[11] F. Dennstädt, J. Hastings, P. M. Putora, M. Schmerder, and N. Cihoric, "Implementing large language models in healthcare while balancing control, collaboration, costs and security," *NPJ Digital Medicine*, vol. 8, no. 1, p. 143, 2025. DOI: 10.1038/s41746-025-01476-7.

[12] D. Falessi et al., "Empirical software engineering experts on the use of students and professionals in experiments," *Empirical Software Engineering*, vol. 23, no. 1, pp. 452–489, 2018. DOI: 10.1007/s10664-017-9523-3.

[13] S. Boguslawski, R. Deer, and M. G. Dawson, "Programming education and learner motivation in the age of generative AI: Student and educator perspectives," *Information and Learning Sciences*, vol. 126, no. 1/2, pp. 91–109, 2025. DOI: 10.1108/ILS-10-2023-0163.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[14] M. Krishna, B. Gaur, A. Verma, and P. Jalote, "Using LLMs in software requirements specifications: An empirical evaluation," in *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, IEEE, Jun. 2024, pp. 475–483. DOI: 10.1109/RE59067.2024.00056.

[15] M. A. Umar and K. Lano, "Advances in automated support for requirements engineering: A systematic literature review," *Requirements Engineering*, vol. 29, no. 2, pp. 177–207, 2024. DOI: 10.1007/s00766-023-00411-0.

[16] K. Ronanki, C. Berger, and J. Horkoff, "Investigating ChatGPT's potential to assist in requirements elicitation processes," in *2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE, Sep. 2023, pp. 354–361. DOI: 10.1109/SEAA60479.2023.00061.

[17] N. Perry, M. Srivastava, D. Kumar, and D. Boneh, "Do users write more insecure code with AI assistants?" *CCS 2023 - Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2785–2799, Dec. 2023. DOI: 10.1145/3576915.36231.

[18] X. Zhu et al., "When software security meets large language models: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, no. 2, pp. 317–334, 2025. DOI: 10.1109/JAS.2024.124971.

[19] H. Cheng et al., "Generative AI for requirements engineering: A systematic literature review," *Software: Practice and Experience*, 2025. DOI: 10.1002/spe.70029.

[20] M. S. Jalali, M. Siegel, and S. Madnick, "Decision-making and biases in cybersecurity capability development: Evidence from a simulation game experiment," *The Journal of Strategic Information Systems*, vol. 28, pp. 66–82, 1 Mar. 2019. DOI: 10.1016/j.jsis.2018.09.003.

[21] S. Baltes and S. Diehl, "Towards a theory of software development expertise," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ACM, Oct. 2018, pp. 187–200, ISBN: 9781450355735. DOI: 10.1145/3236024.3236061.

[22] S. L. Vadlamani and O. Baysal, "Studying software developer expertise and contributions in Stack Overflow and GitHub," in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, IEEE, Sep. 2020, pp. 312–323. DOI: 10.1109/ICSME46990.2020.00038.

[23] B. Jelen et al., "Some experience required: Comparing professional requirements engineering experience to students' perspectives," in *2025 IEEE/ACM 37th International Conference on Software Engineering Education and Training (CSEE&T)*, IEEE, 2025, pp. 303–313. DOI: 10.1109/CSEET66350.2025.00038.

[24] V. B. Kampenes, T. Dybå, J. E. Hannay, and D. I. Sjøberg, "A systematic review of quasi-experiments in software engineering," *Information and Software Technology*, vol. 51, no. 1, pp. 71–82, 2009. DOI: 10.1016/j.infsof.2008.04.006.

[25] M. Poženel, L. Fürst, D. Vavpotič, and T. Hovelja, "Agile effort estimation: Comparing the accuracy and efficiency of planning poker, bucket system, and affinity estimation methods," *International Journal of Software Engineering and Knowledge Engineering*, vol. 33, no. 11n12, pp. 1923–1950, 2023. DOI: 10.1142/S021819402350064X.

[26] E. Miranda, "Moscow rules: A quantitative exposé," in *International Conference on Agile Software Development*, Springer, 2022, pp. 19–34. DOI: 10.1007/978-3-031-08169-9_2.

[27] D. Fujs, S. Vrhovec, and D. Vavpotič, "Balancing software and training requirements for information security," *Computers & security*, vol. 134, p. 103 467, 2023. DOI: 10.1016/j.cose.2023.103467.

[28] OWASP Foundation, "Owasp Application Security Verification Standard (ASVS)," Version 5.0.0, 2025, Accessed: Jan. 23, 2026. [Online]. Available: https://owasp.org/www-project-application-security-verification-standard/.

[29] National Institute of Standards and Technology (NIST), "Nist risk management framework (RMF)," Created November 30, 2016, Updated September 10, 2025, 2025, Accessed: Jan. 23, 2026. [Online]. Available: https://csrc.nist.gov/Projects/risk-management/about-rmf.

[30] K. Okoye and S. Hosseini, "Mann–Whitney U Test and Kruskal–Wallis H Test statistics in R," in *R programming: Statistical data analysis in research*, Springer, 2024, pp. 225–246. DOI: 10.1007/978-981-97-3385-9_11.

[31] D. Chicco, A. Sichenze, and G. Jurman, "A simple guide to the use of Student's t-test, Mann-Whitney U test, Chi-squared test, and Kruskal-Wallis test in biostatistics," *BioData Mining*, vol. 18, no. 1, p. 56, 2025. DOI: 10.1186/s13040-025-00465-6.

[32] L. Pavlič, V. Saklamaeva, and T. Beranič, "Can large-language models replace humans in agile effort estimation? lessons from a controlled experiment," *Applied Sciences*, vol. 14, no. 24, p. 12 006, 2024. DOI: 10.3390/app142412006.

[33] S.-F. Wen and B. Katt, "A quantitative security evaluation and analysis model for web applications based on OWASP application security verification standard," *Computers & Security*, vol. 135, p. 103 532, 2023. DOI: 10.1016/j.cose.2023.103532.

[34] F. M. J. Teichmann, "Platform governance under NIS2 and the cyber resilience act: Cybersecurity by design as social practice," *Information, Communication & Society*, pp. 1–14, 2026. DOI: 10.1080/1369118X.2025.2609780.

[35] K. Khadka and A. B. Ullah, "Human factors in cybersecurity: An interdisciplinary review and framework proposal," *International Journal of Information Security*, vol. 24, no. 3, pp. 1–13, 2025. DOI: 10.1007/s10207-025-01032-0.

[36] A. Pollini et al., "Leveraging human factors in cybersecurity: An integrated methodological approach," *Cognition, Technology & Work*, vol. 24, no. 2, pp. 371–390, 2022. DOI: 10.1007/s10111-021-00683-y.

[37] I. Corradini and E. Nardelli, "Building organizational risk culture in cyber security: The role of human factors," in *International Conference on Applied Human Factors and Ergonomics*, Springer, 2018, pp. 193–202. DOI: 10.1007/978-3-319-94782-2_19.

[38] T. McEvoy and S. Kowalski, "Deriving cyber security risks from human and organizational factors–a socio-technical approach," *Complex Systems Informatics and Modeling Quarterly*, no. 18, pp. 47–64, 2019.

[39] M. Crawford et al., "Redcap to epic interfaces create an efficient hybrid software solution for a public health registry," *Health Informatics Journal*, vol. 30, no. 2, p. 14 604 582 241 249 927, 2024. DOI: 10.1177/1460458224124992.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Obstacles in Implementation, Integration, and Security of Self-Sovereign Identity and Proposed Solutions

Erik Hieta-Aho, Anni Karinsalo, Valtteri Lipiäinen
VTT, Technical Research Centre of Finland
e-mail: {erik.hieta-aho | anni.karinsalo | valtteri.lipiainen}@vtt.fi

***Abstract*—Self-Sovereign Identity is an approach to handling identity documents in a user-centric manner around user consent. Specifically, identity documents are always controlled by the user they identify, and any operations involving them are carried out by the user. This is in contrast to existing models of identity management where information is stored and processed in a centralized manner by an external party. This approach gives rise to many applications including service authentication, cross-border travel, education-focused credentials, and other user-centric technological use cases. However, work is still required for these applications to become reality. One of the key issues is that a fragmented standards space hinders efficient product building. A potential consequence is the emergence of compartmentalized ecosystems, which would drastically decrease systems' user-friendliness. In this paper, we define several Self-Sovereign Identity implementation choices within identity layer, software layer and cryptographic layer. We also explain the security effect of the various implementation choices. We also analyse the impact of obstacles faced in implementing and integrating Self-Sovereign Identity on security and the user experience. Further, we suggest how the situation could be improved, including detailed suggestions for improving the use of scope-defined presentation requests.**

Keywords-Digital credentials; Identification; Decentralized; Standards; Interoperability; Security Profile; Self-Sovereign Identity.

I. Introduction

This document is an extended version of the paper [1] published at the CENTRIC 2025 conference.

As online services become more and more integrated into daily life, concerns about security and privacy become more relevant. More often than not, a service requires authentication, be it for access to paid services or simply access to the history and data of the user. The traditional way of handling authentication, usernames and passwords, has serious flaws: having to use multiple passwords either leads to repetition (which poses a security threat) or makes accessing services cumbersome. Single Sign-On (SSO) offers a solution to this issue: a user only has to remember one password, and the corresponding identity can be used to authenticate to multiple services. A number of standards have been established in this area, including OpenID Connect. However, this approach still means that user data is stored by a centralized authority.

Self-Sovereign Identity (SSI) aims to improve on the SSO approach by making the user (whose identity data is being handled) sovereign over that data. Technically, this means authenticating using Verifiable Credentials (VCs), which contain user information. The credentials are stored by the user and only used at the user's discretion. In addition to plain user data, the credentials can also contain other sensitive information, such as travel documents, without the user having to relinquish control of them to any identity provider. This opens up the possibility for further applications in a privacy-preserving manner.

A natural question is how one can prove their identity in this setup, where the credentials are controlled by the user. The answer is twofold. First, VCs identify their owner by use of a Decentralized Identifier (DID). A DID allows the creation of a persistent identity that is not linked to any centralized authority. There are multiple different DID methods for achieving this goal, from traditional public-key cryptography to blockchain-based solutions. Second, trusted issuers can vouch for the identity of a user. For instance, a government will only issue a passport VC to a user it has identified as the one holding the given passport. In comparison to other digital identity frameworks (e.g., government-issued eID solutions) SSI has the benefit of being decentralized and so the interoperability across borders may be considered less challenging.

A key aspect of the trust that is placed upon digital credentials is ensured by the cryptography that is implemented to sign the credentials. The traditional cryptography that we have trusted for decades and is currently implemented in digital credentials is elliptic curve digital signature algorithms (ECDSA) [2]. Unfortunately due to the accelerating threat of quantum computers and the ability to use Shor's Algorithm to break ECDSA, there is a requirement to transition to a new family of algorithms known as Post Quantum Cryptography (PQC). Thankfully the National Institute of Standards and Technology (NIST) in the USA put out a call in 2016 for new cryptography algorithms that would be considered secure against quantum computers (and traditional computing) [3]. In 2024, NIST standardized the first set of PQC algorithms, FIPS 203 [4] and FIPS 204 [5]. These algorithms are based on lattice based cryptography. FIPS 204 is the new digital signature algorithm also known as ML-DSA. ML-DSA is the most likely candidate to be implemented to secure digital credentials. While these new algorithms are significantly different in many aspects there has been evidence that ML-DSA performs quite well in securing verifiable credentials.

The SSI approach is still relatively new, and the ecosystem is still taking shape. The unique benefits SSI can bring for users are only achievable if systems are secure, as described in Section IV, and interoperable, as described in Section II-B. For this reason, it is crucial to understand the several

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

implementation choices, as described in Section III. Standards are a great starting point for interoperability, fortunately there is a rich standards ecosystem for SSI, as described in Section V. Even though standards exist in this space, there are major areas of friction when attempting to integrate different systems, as we have faced in a project implementing SSI for authentication. The challenges mainly relate to the flexibility in encoding formats, and are described in detail in Section VI. We believe that these obstacles may cause major issues for the establishment of an SSI ecosystem, and in Section VII we describe approaches, mostly related to clarity in metadata and standards texts, for lessening this impact.

A. Related work

While many works in this area list some challenges, there are relatively few that put a specific focus on identifying and analyzing obstacles in the up-take of SSI.

The work [6] gives a broad survey of SSI technology, and proposes a taxonomy for both the architecture of SSI systems and challenges facing them. The challenges presented land in five categories. The first category relates to the layer of 'public utilities', namely, when deploying blockchain technologies as a verifiable data registry. The challenges in this layer are thus due to inherent weaknesses of blockchains, such as the commonly recognized blockchain issue of scalability. Second challenge category relates to the obstacles occurring in 'communication and interface layer': usage of DIDs, recovery of keys and connectivity challenges. Third challenge category relates to the 'credential layer': usage of VCs, whereas the fourth category illustrates application usage-originated challenges in the 'application ecosystem layer', such as lack of tools or frameworks. Finally, the fifth category depicts challenges related to the 'governance layer'. This work is based on a literature review, and as such has a more high-level view of SSI systems. In comparison we provide an implementation-level perspective on challenges faced, in contrast to the taxonomy approach taken by [6]. Our work can be viewed as a detailed analysis of issues at the "credentials" and "communications and interfaces" layers.

The work [7] provides a broad survey of VC and DID technologies beyond just their use in SSI. It also provides a list of challenges in use of VC/DID technologies. Challenges are classified into standardization, scalability, usability, integrity, and security & privacy. The specific challenges include some of the challenges presented by [6]. While valuable, [7] deals with higher-level concerns than what are focused upon within this paper.

Other references that are relevant to the SSI ecosystem include: [8] provides a systematization of knowledge of specific security issues within SSI; [9] provides a further review and taxonomy of SSI; and a fuller list of existing surveys can be found in [7].

II. SSI

The original concept for SSI is defined by [10] and further refined by [11]. Essentially, the following principles were defined:

- Existence: Users must have an independent existence beyond digital systems.
- Control: Users must control their identities and how they are used.
- Access: Users must have access to their own data without intermediaries.
- Transparency: Systems and algorithms managing identities must be open and transparent.
- Persistence: Identities should be long-lived and ideally last indefinitely.
- Portability: Identities should be transportable across different systems.
- Interoperability: Identities should work across various platforms and technologies.
- Consent: Users must agree to how their identity data is shared and used.
- Minimization: Identity systems should only collect and share necessary data.
- Protection: Users' rights and privacy must be safeguarded.

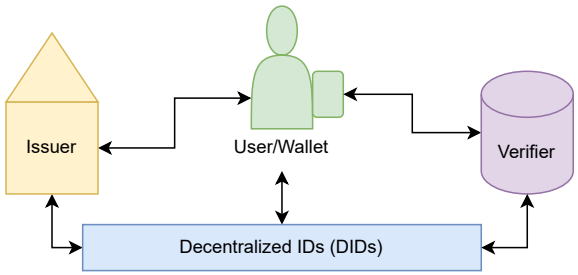


Figure 1. Self-Sovereign Identity

To fulfill this promise, a set of technologies have been proposed. For handling identity, the technologies of decentralized identifiers and verifiable credentials can be used.

VCs are objects that encode verifiable information about a user. Fundamental to the SSI approach is a separation of the actors into three distinct groups: wallets, issuers, and verifiers, which can be seen in Figure 1. The wallet is where a user stores their credential. The issuer is the party that issues credentials to be stored in the user-wallets. The verifier is then in charge of checking whether a given credential is valid. All three have to interact for the system to work.

An example of this setup in action is a university diploma and can be seen in Figure 2. After a student graduates, they may want to receive a digital diploma, which they can use in the future to prove they have graduated (for instance when applying for jobs, or applying for a further degree). Following an SSI approach, the student should wholly control how they use the diploma after it is granted: there should be no need to go back to the university each time (and for instance let them know what jobs the student is applying to). For this reason, the student should have an SSI wallet, which is independent of the university.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

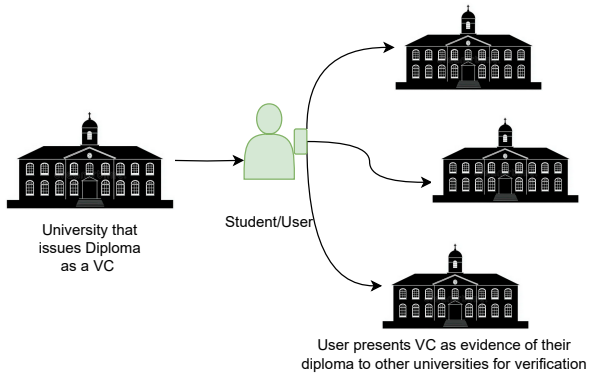


Figure 2. User-centric University Credentials

To receive a university diploma VC, the student would then interact with an issuer tied to the university, the party that is authorized to issue such diplomas. After the student has shown that they are indeed entitled to a credential, the issuer creates and signs a credential tied to that student (through their unique DID), and sends it to the user. After this point, the credential is controlled by the student. For the credential to be useful, anyone it is presented to should be able to confirm it was issued by the university to this person. This is where the verifier becomes relevant. They are in charge of verifying that the credential is valid, and that this student has indeed graduated from the university.

This sort of setup is useful for making it clear how the roles of the wallet, issuer and verifier are different entities. There might also be setups where the issuer and verifier are the same entity, for instance if the credential in question is used to log in to only one service. In the SSO use-case, we could also have a setup where one entity issues credentials that are accepted for login purposes by multiple different services. It is clear that this setup requires that the three parties (the wallet, issuer, and verifier) have ways of communicating with each other.

There are two core security and privacy issues to take into account here. First, it should be impossible for anyone to convince a verifier they hold a credential when they don't. Second, the user should be able to keep their personal data private for as long as they prefer.

For the first point, cryptographic tools, such as digital signature algorithms are used. Each credential is signed by the issuer, and this signature is what allows the verifier to determine that the credential indeed has been issued.

For the second, the decentralized approach goes a long way: the user only has to interact with the issuer once. There are functionalities that go further than this, for instance zero-knowledge proof based approaches, which allow the user to only reveal part of the information in the credential.

Various practical use cases exist for SSI in the academic literature, namely for blockchain-based identity and access management systems for IoT focusing on smart vehicles [12], addressing banking challenges in Know Your Customer (KYC) process [13], mapping SSI concepts to the healthcare domain [14], application of SSI in event ticketing systems [15], mobile SSI access control systems enabling secure peer-to-peer communication without internet connectivity [16], facilitating secure identity verification for decentralized energy systems [17], digital identity management [18] and facilitating border control access and travel data inspections [19].

For these use cases, SSI provides industry-specific benefits: in IoT, SSI can improve autonomy and resilience of the devices, whereas for the banking sector, SSI enables compliance adherence and efficiency of operations. For healthcare, an essential attribute is naturally privacy, for which SSI can be used to improve. In ticketing and related recreational applications, SSI can prevent fraud, and for access control, SSI's benefit is to maintain resilience.

A. SSI as a user-centric technology

As described previously, SSI is at its core a more user-centric version of existing SSO solutions. This solution is more user-focused in three main ways: it allows the user flexibility, it prioritizes user control over their data, and it enables them to protect their privacy.

Due to the decentralized setup, the system provides flexibility to users. They can choose from a number of wallet applications, which may have different features, but are still able to receive and use credentials from any issuer.

SSI prioritizes user control over their personal data. Credentials are stored in the user-wallet, and only shown to others when the user expressly consents to. On a technical level, this is achieved by using a *self-issued OpenID provider*. The user is able to make claims about themselves (possibly leveraging credentials they have previously received) without having to pass through an OpenID provider as is traditionally done in SSO setups.

Other functionalities can protect user privacy even further. Approaches based on zero-knowledge proofs can allow users to only reveal a subset of the information contained in their credentials, and wallet applications can keep track of the information revealed by users to ensure that not too much is revealed to any one party. Overall, the decentralized approach of multiple interoperable wallets allows wallet applications to cater to user needs both in privacy and for general user interface behaviour.

In terms of identity revocation in SSI, standards, such as [20], do not define specific revocation format or method. There exist several approaches to implement revocation in SSI, but a common system-wide challenge is that the mechanisms usually rely on some kind of centralization, which is in conflict with the basic principles of SSI [21].

B. The need for interoperability

As put forth by the original proposers of SSI, identities should be interoperable. This is important for a fundamental reason (identities as commonly understood are not tied to any one system), but is also necessary for a technical reason. If

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

identities are user-controlled, they have to also be decentralized. A decentralized system collapses back to centralization if it is not possible for different systems to work together.

Due to the high level of effort necessary to receive documents containing sensitive personal data, user adoption is dependent on the systems to be easy to use after setup. If a single SSI wallet does not work across systems, using SSI becomes much more cumbersome.

Further, for wide adoption (which is necessary for a true digital identity) it is necessary for a large number of service providers to support authentication using SSI. If each provider has to implement the entire SSI stack for themselves, this will likely create a major obstacle. Therefore, it is important for new players to be able to rely on the existing infrastructure. This becomes much harder if the systems are not interoperable.

III. Implementing SSI

In this section we discuss the technical choices that should be considered when setting up an SSI stack. However, we exclude broad descriptions regarding trust governance and liability management of the ecosystem. Similarly, we exclude the description regarding compliance with regulation.

A. Identity layer

Identity layer includes how identity is represented and properties of root of trust for the identity. More specifically, these relate to choices for identity model, DID methods, type of verifiable data registry and verifiable credential. This layer requires answers to the following questions: who is the identity holder? Does the VC need selective or ZKP functionality? What properties are needed from the verifiable data registry, for instance in terms of revocation method? Who governs the registry?

The chosen identity model describes whether the used identity is for a natural person, a device or for instance an organization. This affects for instance the selection of a credential type and the identity proofing process.

DID methods are the mechanisms with which the DID and DID document are created, resolved, activated and deactivated [22]. In other words, they include guidance of how the identity that they refer to can be used. In practice, the DID method also determines how trust is anchored: for example, through a public blockchain, a permissioned ledger, a web domain, or another registry mechanism. There are numerous DID methods available, and choosing a DID method type affects the system properties fundamentally. It affects the availability and performance of DID resolution, the degree of decentralization, the privacy of metadata publication, and system dependence on external infrastructure.

Verifiable data registry type is used to publish and resolve identity-related metadata. The choice of verifiable data registry type affects system properties such as security, performance and scalability. For instance, increasing registry's decentralization increases transparency and resilience but it also increases latency and very likely costs. Verifiable data registry can be something like a public or permissioned blockchain, on the other hand it could be a governmental managed central registry. The registry choice also affects how credential status information can be maintained or revoked. Thus, implementing a registry requires considering a revocation mechanism: how and with which conditions the credential remains valid. Basically, implementations for revocation of identity are often based on the revocation lists similar to those familiar to Public Key Infrastructures (PKIs), often combining cryptographic accumulators with the mechanism. Another path is to build and utilize dependencies either between VCs, or between Issuer and Verifier, in which VCs validity can be limited for example by time or other parameter(s).

Another identity management-related choice is the selection of where identity related functions are executed and where the keys will be stored. These functionalities can be implemented either on-device, on server side or as a hybrid version for which there is e.g., a key backup or complementing functionality in cloud or a separate device.

B. Software stack

SSI software stack layer includes several software components that interact with each other and through which SSI functions are realized in practice. Deployments depend on the interaction between wallets, issuer systems, verifiers, registries, agents and their APIs, as well as the frameworks that ensure trust. The purpose and role for each of these have to be defined. For example, the degree of authorization services that is required from the issuer has to be defined: whether the issuer will handle the authorization or there would be a separate agent managing this.

C. Cryptographic layer

Cryptographic layer relates to choices regarding implemented signature schemes and other cryptographic mechanisms in the SSI system. This layer requires answers to the following questions: What kind of VCs are used? Does the system have to be quantum secure? What kind of interoperability should the schemes and credentials allow? What are the cryptographic requirements in terms of revocation choices? There are several signature schemes depending whether we want to use quantum secure or traditional signatures. This affects, based on chosen scheme, not only the security properties but also verification time and interoperability capabilities. The objectives of the identity layer affect the used credential formats. If the system should support for instance selective disclosure, the credential format should support it. When selecting credential formats, a choice can be made between a credential representation, for example JWT-based credentials, integrity proofs such as JSON-LD or selective disclosure mechanisms such as SD-JWT [23]. From the selective mechanisms, a choice can be made to allow a subset, or no information disclosed (ZKPs). Cryptographic layer also defines the used credential status and revocation mechanisms. Status verification functionalities may require techniques such as status lists or cryptographic accumulators.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

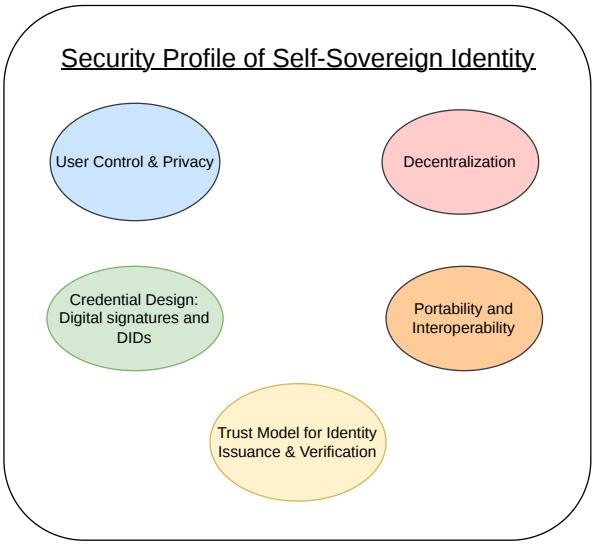


Figure 3. Security Profile of Self-Sovereign Identity

IV. Security profile of SSI

The trust and security that SSI and digital credentials are built upon is essential for the ecosystem to thrive and is visualized in Figure 3. There are a few pillars that ensure the security of an SSI framework. Firstly the communication and interaction between three key components, the wallet, the verifier, and the issuers is absolutely required to be standardized and designed to be interoperable. Along with each of these three components comes the required trust for each of them to be secure and trustworthy. Furthermore, the DID methods and design of the verifiable credentials, including the cryptographic algorithms that are implemented to secure the credential signatures, is essential for the trust of the system. There is one more aspect of the SSI framework that allows for the digital framework to function and that is the data registry, and this aspect is what allows for the decentralized construction of the framework.

A. Ecosystem Trust Model

The design of the trust model has a large impact on the software stack and security model that one implements. As the identity layer and trust model determine the requirements of the SSI ecosystem, they establish the technical needs that must be fulfilled.

While the cryptographic security of digital credentials is guaranteed (as long as it is properly implemented) this does not necessarily ensure trust. The trust of the ecosystem must be built into its foundation. Therefore, it is especially important to ensure the integrity of each of the components and all of the actors taking part in the ecosystem.

For instance, the entities that will be issuing the digital credentials have a significant impact upon the trust of the architecture. As such it is important to have a method for ensuring the legitimacy of issuers in the SSI ecosystem. As described in [24] the implementation of an Issuer Identity Registry is one method of ensuring the trust in the entities that issue the credentials. While the issuer registry does not create credentials, it does create the trust in the organizations that are issuing the credentials to the end users.

Furthermore, another decision that is important for ensuring trust in the ecosystem includes the design of the verifiable data registry (VDR) [25]. While again the digital credentials are secured by their cryptographic layer, the storage and maintenance of the VDR has a large impact on the trust model. The core functionalities of the VDR include identifier management, trust anchors, and revocation management. The VDR stores and manages the DIDs and the DID documents that relate to each of the VCs. The trust anchors provide access to the public keys associated to the issuers and the credential schemes that allow for verification that the claims are issued by the trusted entity.

B. Ensuring a Secure Software Stack

The software stack covers the three core components (wallet, issuer, and verifier) and the data registry (storage of DIDs). The impact on the security of the SSI framework is ensured by the proper implementation and development of these four components. There are a variety of decisions that must be made when constructing a secure framework and each of the various decisions impacts the security profile of the entire system.

The choice of the three core components must be considered carefully to ensure their interoperability. The decisions made in the identity layer and trust model are what then determine the properties and expectations of the verifiable credentials which then establish what is required of the software stack.

The VDR is the component that includes the storage of public key cryptographic material that allows for flexibility of the framework to be decentralized or centralized, depending on the particular needs of the framework. There are a variety of options that could be used to demonstrate the decentralized model of SSI, two main options are distributed ledgers and decentralized databases [26]. The centralized models would include a centralized trusted database such as trusted government ID databases.

C. Cryptography

The cryptographic layer of SSI is heavily dependent upon the public key cryptography algorithms known as digital signatures. These digital signatures are the secure method for authentication and verification of the credentials. In current implementations the security of the credentials is typically secured by implementation of elliptic curve digital signatures (ECDSA).

D. Future cryptography changes and impact

The digital signatures that are implemented to secure the verifiable credentials in an SSI stack are required to be changed by 2030. As one would consider the security of ones identity data to be critical, the usage of the SSI credentials would be considered part of critical infrastructure.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

As the new PQC algorithms are significantly different than the traditional algorithms, it is important for there to be standards by which entities can follow to ensure successful interoperability of the SSI ecosystem.

V. Current standards landscape

Standards are a major enabler of interoperability. They provide a shared reference for providers to use. Fortunately, SSI already has well-established standards. Roughly, the standards can be broken down into four layers: technical, credentials, identity, and institutional. See Table I for a list of standards.

On the technical level, standards describe the way data is encoded. IETF has standardized JSON Web Tokens & Signatures (JWT & JWS) [27], which can be used to encode verifiable credentials. A different format, JSON-LD, has been standardized by W3C. The multicodec protocol, partly standardized by W3C, can be used to flexibly encode key material, which is necessary when defining DIDs.

These standards don't yet define the objects relevant to SSI. The necessary standards for SSI are defined by W3C, which standardize the core verifiable credentials data model [20], as well as various extensions of it. When combined with standardized encoding formats, it is then possible for parties to store, send and process verifiable credentials in a standardized way. In addition to defining the credential, these standards also define verifiable presentations, which is a format credentials are converted to in order to be presented to a verifier.

To be able to use these credentials, standards are still needed to describe how parties exchanging the credentials should operate. Here is where the OpenID Foundation standards [28] come into play. The protocols for both issuing and verifying credentials are heavily based on existing OpenID standards. The SSI-specific ones include OpenID for Verifiable Credential Issuance and OpenID for Verifiable Presentations. These define OpenID-based flows for a user to receive a credential (after authentication), and for authenticating using a verifiable presentation in a way which is interoperable with existing OpenID-flows.

The above flows leave some details open in the interest of future flexibility. This creates an issue for institutional players that would like to support the use of SSI. Consequently, various institutional players have stepped in to further specify interoperability conditions in given context. In the EU, EBSI [29] has defined a set of conformance tests for software, which specify the interaction between the components, and the EU digital identity wallet places requirements for member states in the EU [30]. In the private sector, organizations like the International Data Spaces Association [31] and GAIA-X [32] define SSI used together with data spaces.

VI. Obstacles to integration

As laid out in Section III, interoperability is necessary for self-sovereign identity to become a reality. This requires integration between various components. However, there are multiple obstacles to this integration work. In this section, we point to places where the friction is especially high. Many of these have to do with incompatible formats.

The issues in this section were identified during the Horizon Europe project TANGO [33]. In that project, SSI solutions were piloted for authentication purposes in a data spaces settings. This requires interoperability between multiple parties, including the providers of the components in the data space, as well as providers of open source software used for SSI functionality. In that context, interoperability issues led to increased developer time spent on troubleshooting and working out how to resolve points of incompatibility. Where interoperability issues could not be completely resolved, this led to extra complication in system setup, as well as friction in the user experience. The issues as well as their impact are discussed in detail below. See Figure 4 for an overview.

These obstacles cause various issues both in development and in user experience. As such, they create friction that results in

1) Friction when integrating systems, slowing down development work.
2) Technical impossibility, given time constraints, of making two systems interoperable.
3) Making projects harder to maintain due to needing code purely for compatibility reasons.
4) End users not being able to use their credentials in different systems, putting in question the amount of sovereignty they have over their identity data.
5) Security issues caused by software being more complicated than necessary, due to a need to bridge between systems.

Overall, these issues reduce innovation by slowing down development work, and may lead to the establishment of non-interoperable ecosystems. For instance, friction caused by unclear formats might cause significant resources to be spent troubleshooting the issue.

A. VC format

The data model of a verifiable credential is defined by the W3C [20]. However, the data model does not define how exactly the credential will be encoded. The two major encodings in use, JWT and JSON-LD, are both JSON-based. However, they have major differences. The JSON Web Token (JWT) format stores credential information in a JSON object, which is then encoded in base 64 with a signature (in JWK form) appended. On the other hand, JSON with Linked Data (JSON-LD) supports various cryptographic suites for verifying the authenticity of a credential, and does not have a standard base64 encoding.

The issue is that a software component that supports credentials in one format will not automatically work for the other. As JSON-LD is more flexible in its cryptographic functionality, advanced functionality (such as ZKP-based selective disclosure) can be implemented more easily with this encoding. These advanced functionalities will fail to function if any part of the stack does not support the JSON-LD encoding of a credential

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. STANDARDS AND ENTITIES RELEVANT FOR SSI

Standard	Layer	Role	Standards body or Entity
JOSE	Technical	Defines basic data objects	IETF
JSON-LD	Technical	Format for data objects	W3C
multicodec	Technical	Encodes key data	W3C
Verifiable Credentials Data Model	Credentials	Defines basic credential and presentation data formats	W3C
OIDC	Identity		OpenID Foundation
EBSI	Institutional	European Blockchain Services Infrastructure	European Commission
DID	Identity		W3C
EU digital identity wallet	Institutional		European Commission
IDSA	Institutional	Standards organization	International Data Spaces Association
GAIA-X	Institutional		GAIA-X Association

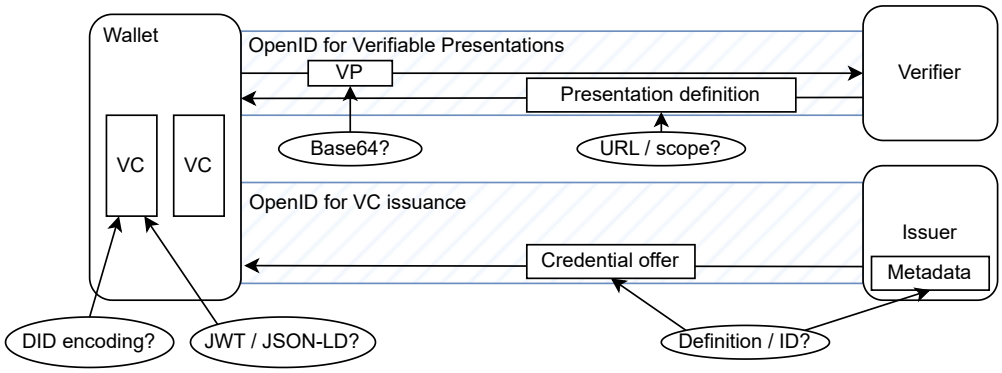


Figure 4. Diagram of problematic ambiguities in SSI flows. Each oval label identifies an area of divergence across implementations. See Section VII for detailed explanations.

Unless additional work is carried out by the software provider (an issue in its own right, especially for an emerging technology), an end user will encounter situations where their SSI wallet fails. This is especially true if advanced functionality is used, which may have been a reason a user adopted an SSI wallet. If two different providers use different formats, the user might well miss out on functionality important to them, for instance the use of zero-knowledge-proof enabled credentials. Further, they may need different versions of the *same* credential in different contexts, making the setup cumbersome and less intuitive.

B. DID key encodings

With some DID methods, especially `did:key`, there is a need to encode a public key. This public key is to be read by the verifier when assessing whether the credential is valid. However, there are multiple different encoding formats for the key, even when the underlying algorithm is the same. Even though the multicodec format provides a method for determining the encoding used, it does not guarantee that support for any one encoding is present.

This variation in the possible encodings means that developers have to support multiple different encodings, risk hard-to-debug errors, or lead to a fracturing of the ecosystem based on the encoding types supported.

From the user point of view, in addition to situations where a credential fails to validate, the encoding differences give rise to issues related to identifiers. In the DID scheme for defining identifiers, the public key (in some specific encoding) is used as part of the identifier. As such, requiring credentials with different DID encodings results in credentials belonging to the same person having to be issued under different identifiers. This complicates the landscape for the end user, and takes away from the goal of SSI principle of “existence” and “transparency”.

C. Presentation format

In order to use a verifiable credential, it needs to be converted to a verifiable presentation. In the `vp_token` flow, this verifiable presentation is then sent to the verifier for checking. However, the exact format this presentation takes is not defined in the relevant standard [34], and can differ across implementations in practice. It can be sent as a JWT, or that JWT can be encoded as a base64 string (which further can differ, e.g., whether it is URL-safe, and how padding is handled).

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

These differences mean that parsing might fail if a different-than-expected encoding is used. The resulting error can be hard to debug, since failure of a presentation is an expected scenario (in the case a user has not presented a correct credential), and it can be good security practice to not reveal exactly what was the reason for rejecting an authentication attempt.

Again, this means that a user-wallet might fail to function correctly for unclear reasons, which also leads to development time being spent on troubleshooting.

D. Breaking metadata changes

During the issuance flow, the credential issuer needs to expose some metadata about itself. An important piece of information contained in the metadata is the types of credentials an issuer offers. However, there have been changes to this mechanism in the drafts of OpenID for Verifiable Credential Issuance standard. In older versions, this information was contained in the credential offer itself, but later versions opt for an approach where the credentials offered are defined in the metadata of the issuer, and the credential offer identifies the credential being offered via an ID.

This change creates challenges especially in wallets that follow a frontend-backend architecture. In the first approach, the credential offer contains everything needed to describe the credential to the user. In the second, fetching of the information becomes more complicated, since the metadata of the issuer has to be specifically accessed.

For the user, this sort of change can lead to sudden issues with issuance even with issuers that worked previously.

E. Scope vs presentation definitions

When a verifier describes the type of credentials it is expecting, it has various different ways of doing this. A `presentation_definition` describes the contents expected of the credential. However, the verifier can also make use of a `scope`, which is an arbitrary string that refers to some credential ("Either a dcql_query or a scope parameter representing a DCQL Query MUST be present in the Authorization Request, but not both" [34, section 5.5] [28]). However, this "scope" is hard for a wallet to be interoperable with, since the vocabulary is not defined in standards: "The specific scope values, and the mapping between a certain scope value and the respective DCQL query, are out of scope of this specification." [34, section 5.5]. Since the mapping is not defined, an implementation has no recourse when met with an unknown scope value. To support a verifier that describes credentials using a scope, it is necessary to first define the meanings of the scopes between the organizations providing the wallet and verifier, and second for the wallet to specifically implement these scopes to properly support them.

Because of this, when processing a credential presentation definition, it is not certain that parsing will be possible, due to the use of a (possibly) nondefined scope.

This leads to various issues for both end users and developers. Extensive work is required to support all scope values a user-wallet might end up being requested, as well as handling situations where an unknown scope is used. Further, discussion might be necessary between organizations serving as verifiers and as wallet providers in order to define the scope values in use. From the end user perspective, if an unknown scope is used, they may not have information about what credential is being requested. As happened in the TANGO pilots, a user may be required to choose the credential to present without having solid information on what is being requested. In the worst case, this might lead to unintended disclosure of credentials, going against the principle of minimization.

VII. Solutions: what would help

The previous section outlined 5 integration obstacles observed when attempting to implement an SSI system in practice, as well as the impact those issues have. This section contains proposals for changes in the landscape that would make integration work more seamless, and make it more likely for SSI to be used widely.

There are two main actors who can help integration work: standardization bodies and technical specifications. The solutions suggested in this section apply to both, as well as the *drafts* of standards that are used to build interoperable SSI systems before final standards are published.

In all of the solutions suggested below, there is a trade-off between adding complexity and easing interoperability. However, the cost of the changes proposed is relatively low, and all technical changes could be intended as optional additions. This would not complicate simple in-house setups, but would provide useful mechanisms for avoiding interoperability issues and associated work when developing solutions meant to work across organization. The goal of interoperability is central for SSI to fulfill its principles, and ecosystem fragmentation must be prevented for it to become a reality.

See Table VII for a summary of the solutions suggested and the issues they address. The solutions can be roughly split in three categories:

1) Providing machine-readable information about implementations. These would help develop software that is robust against differences across implementations, at least providing clear feedback on what interoperability issues might be caused by.
2) Changes to the standards and specifications themselves to avoid ambiguities when producing implementations.
3) Mechanisms for decreasing ambiguity around the `scope` parameter for defining presentation requests.

Solution 1

Standard and draft versions should be included in machine-readable metadata of applications implementing SSI.

A major hurdle is uncertainty over the draft version of standard implemented by any given application. This can break interoperability either since there are breaking changes,

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Issue	Mitigation
VC format differences	Clear interoperability information in standards and requirements for privacy functionality (Solutions 3 – 5)
DID key	Clear interoperability information in standards (Solutions 3 – 4)
Presentation format	Clear interoperability information in standards (Solutions 3 – 4)
Breaking changes	Draft version metadata and breaking changes warnings (Solutions 1 – 2)
Scope vs presentation definitions	Mechanisms for finding scope definitions (Solutions 6 – 8)

or because a newer draft includes a new functionality. The draft version being clearly communicated in machine-readable metadata would make things smoother. Software could robustly pick the draft version to use, or at least give a clear indication of the issue (unsupported draft version). This would make it much easier for actors to develop systems, and stress-test the standard drafts themselves. This is especially relevant in the draft phase, where changes happen frequently. This and the above change could be implemented simply by including an optional field in issuer and verifier metadata that contains either a single (draft) version, or an array.

Solution 2

A standard draft should specify any significant point which differs from those in a previous version.

This would help application developers update their systems as required. Version history is often provided in standard drafts, but it would be especially helpful if these changes were communicated in machine-readable metadata. This could then potentially be implemented in issuer or verifier metadata similarly to the version number.

Solution 3

Standards and technical specifications should together include exact technical details necessary for interoperability.

Standards support integration work as an intermediary between organizations by providing a common flow for them to agree on. Organizations can ideally build interoperable systems with little or no back-and-forth communication. However, this benefit is greatly reduced if the exact technical details are left out. This can cause issues that are hard to troubleshoot. This can to some extent be covered by standards themselves, but part of the work is left to technical specifications.

Solution 4

Standard drafts and technical specifications should clearly state which details are settled and which are not.

It is not always feasible or desirable to define all technical details. The standard might only be in the drafting phase, where work put into deciding and specifying technical details might be wasted if the final approach ends up being different. The authors of the standard might also want to avoid over-specifying the technical solution. This can leave space for innovation, support more use cases, or leave flexibility in case of further changes. For instance, setting in stone specific cryptographic signature algorithms would cause major issues if there were to be a need to change these (as in the case of the transition to post-quantum cryptography), and as such it is common practice to define these outside the standard itself. In any case, such areas where technical details are not (yet) agreed on, this should be made as clear as possible. This way organizations wishing to build interoperable systems have a clear basis for discussions. Further, it would be useful if there were a mechanism for implementations to communicate the specific choices made. Software can then adapt to the specific choices within the implementation.

Solution 5

Requirements related to user privacy should be set clearly by either standards or technical specifications.

SSI enables the use of many privacy-preserving technologies that can help users keep their data private. However, these technologies may require, for instance, the use of a specific format to store a credential, which in turn might not be supported by all wallets, issuers, or verifiers. This puts the user in a difficult position: they have adopted a technology, which promises them privacy, but in practice these functionalities might not be available. If full convergence of the standards is not feasible, actors wanting to support the takeup of SSI should place clear requirements for protection of user privacy, so that users can trust that the promise of user privacy is actually fulfilled.

A. Undefined scope values

As explained in the previous section, while requesting credential presentations using scopes can be justified in some situations, they create issues for both the end user and developers. The root of the issue lies in the fact that the valid scopes are not defined in specifications, and there is no mechanism for determining whether a given credential matches a scope. There are multiple potential solutions to this issue:

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Solution 6

A registry could be set up (officially or unofficially) for widely known scope values. This would allow for querying the presentation definition in cases where scope is used, and assumed to be widely known. Inclusion in such a registry should not be mandated, to allow for use of private scope values.

Solution 7

The scopes supported by a verifier could be included in (optional) metadata of the verifier, together with corresponding presentation definitions.

Solution 8

In case a credential is meant to be used to respond to a specific presentation scope, this fact could be included in the verifiable credential itself.

VIII. Conclusion and further work

This paper is an extension of [1], and as such extends the description of the technical structure of the SSI ecosystem, including the various decisions that are necessary to ensure the secure trust model is implemented correctly. This version also extends the details regarding the challenges in ensuring the interoperability of the SSI ecosystem.

One key aspect to recognize from this paper is the challenges that scope values bring to the interoperability of an SSI ecosystem. Specifically, it is recommended for there to be a mechanism for determining whether a credential matches a scope. While this may be difficult, our recommended solutions could allow for the ecosystem to be interoperable with various tools such as scope-defined presentation values.

SSI is a promising identity technology, which can offer improvements to user experience and privacy. Due to the nature of SSI as a decentralized technology, interoperability is of great importance. Even though there is a robust set of standards for the different aspects of SSI, there are caveats that cause significant friction when integrating SSI systems in practice. We believe there are a number of approaches that could be taken now to make these systems easier to integrate, and consequently to foster further innovation in the space.

As this paper is based on experiences gathered during our participation in the Horizon Europe TANGO project, with one set of software components. Determining if these issues are widespread, and identifying the obstacles faced in other projects could provide valuable input for standardization bodies. Furthermore, collecting feedback from a diverse set of stakeholders, such as wallet developers, end users, or institutional issuers would also be beneficial to the development of the SSI ecosystem. All of their differing perspectives could validate and contextualize the identified issues.

In future work, a cost-benefit analysis of the value brought by flexibility vs the cost imposed on integration efforts might allow for stronger recommendations for clarity in standards than the ones presented here, which defer to the value of flexibility.

Acknowledgment

This report has been funded by the TANGO project, which has received funding from the European Union's HE research and innovation programme under the grant agreement No. 101070052.

References

[1] E. Hieta-aho, V. Lipiäinen, and A. Karinsalo, "Obstacles in implementing and integrating self-sovereign identity and proposed solutions," in *The Eighteenth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC)*, Lisbon, Portugal: IARIA Press, 2025.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997, ISSN: 1095-7111. DOI: 10.1137/s0097539795293172.

[3] National Institute of Standards and Technology, "Post-quantum cryptography," U.S. Department of Commerce, Washington, D.C., Tech. Rep., 2024. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography.

[4] National Institute of Standards and Technology, "Module-lattice-based key-encapsulation mechanism standard," U.S. Department of Commerce, Washington, D.C., Tech. Rep., 2024. DOI: 10.6028/NIST.FIPS.203.

[5] National Institute of Standards and Technology, "Module-lattice-based digital signature standard," U.S. Department of Commerce, Washington, D.C., Tech. Rep., 2024. DOI: 10.6028/NIST.FIPS.204.

[6] A. Satybaldy, M. S. Ferdous, and M. Nowostawski, "A taxonomy of challenges for self-sovereign identity systems," *IEEE Access*, vol. 12, pp. 16 151–16 177, 2024.

[7] C. Mazzocca *et al.*, "A survey on decentralized identifiers and verifiable credentials," *IEEE Communications Surveys & Tutorials*, 2025.

[8] E. Krul, H.-y. Paik, S. Ruj, and S. S. Kanhere, "Sok: Trusting self-sovereign identity," *arXiv preprint arXiv:2404.06729*, 2024.

[9] F. Schardong and R. Custódio, "Self-sovereign identity: A systematic review, mapping and taxonomy," *Sensors*, vol. 22, no. 15, p. 5641, 2022.

[10] C. Allen, "The path to self-sovereign identity," *Life with Alacrity*, 2016.

[11] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," *The Sovrin Foundation*, vol. 29, no. 2016, p. 18, 2016.

[12] M. Naghmouchi, H. Kaffel, and M. Laurent, "An automatized identity and access management system for iot combining self-sovereign identity and smart contracts," *arXiv preprint arXiv:2201.00231*, 2022.

[13] V. Schlatt, J. Sedlmeir, S. Feulner, and N. Urbach, "Designing a framework for digital kyc processes built on blockchain-based self-sovereign identity," *arXiv preprint arXiv:2112.01237*, 2021.

[14] A. Siqueira, A. F. da Conceição, and V. Rocha, "User-centric health data using self-sovereign identities," *arXiv preprint arXiv:2107.13986*, 2021.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[15] S. Ehrlich, V. Schlatt, and N. Urbach, "Exploring the use of self-sovereign identity for event ticketing systems," *Electronic Markets*, vol. 32, no. 4, pp. 1321–1339, 2022.

[16] A. Enge, A. Satybaldy, and M. Nowostawski, "An offline mobile access control system based on self-sovereign identity standards," *Computer Networks*, vol. 219, p. 109 434, 2022, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2022.109434. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622004686 (Retrieved: 8/2025).

[17] V. Gramlich, M.-F. Körner, A. Michaelis, and J. Strüker, "Ssi in the energy sector: A study," *Fraunhofer Institute of Applied Information Technology FIT*, 2023.

[18] R. N. Zaeem *et al.*, "Blockchain-based self-sovereign identity: Survey, requirements, use-cases, and comparative study," *IEEE/WIC/ACM International Conference on Web Intelligence*, 2021. DOI: 10.1145/3486622.3493917.

[19] P. Koskela, A. Karinsalo, J. Paananen, and L. Salmela, "Streamline border control with blockchain towards self-sovereign identity," in *3rd International Conference on Software Engineering, Security and Blockchain (SESBC 2022)*, Dubai, United Arab Emirates: AIRCC Publishing Corporation, 2022.

[20] M. Sporny, G. Cohen, T. T. Jr, M. Jones, and I. Herman, "Verifiable credentials data model v2.0," W3C, Candidate Recommendation, Sep. 2024. [Online]. Available: https://www.w3.org/TR/vc-data-model-2.0/ (Retrieved: 8/2025).

[21] R. Chotkan, J. Decouchant, and J. Pouwelse, "Distributed attestation revocation in self-sovereign identity," in *2022 IEEE 47th Conference on Local Computer Networks (LCN)*, IEEE, 2022, pp. 414–421.

[22] D. Reed *et al.*, "Decentralized identifiers (DIDs) v1.0," W3C, Recommendation, Jul. 2022. [Online]. Available: https://www.w3.org/TR/2022/REC-did-core-20220719/.

[23] D. Fett, K. Yasuda, and B. Campbell, *Selective Disclosure for JSON Web Tokens*, RFC 9901, Nov. 2025. DOI: 10.17487/RFC9901. [Online]. Available: https://www.rfc-editor.org/info/rfc9901.

[24] J. Kitchens, R. Joy, K. Lemoie, R. Schwartz, and G. Walsh, "Issuer identity registry research report," Credential Engine and Digital Credentials Consortium, Tech. Rep., 2025. [Online]. Available: https://credentialengine.org/resources/issuer-identity-registry-research-report-governance-framework/.

[25] M. Sporny, D. Longley, D. Chadwich, and I. Herman, "Verifiable credentials data model v2.0, verifiable data registry," W3C, Tech. Rep., 2025. [Online]. Available: https://www.w3.org/TR/vc-data-model/#dfn-verifiable-data-registries.

[26] M. Alizadeh, K. Andersson, and O. Schelén, "Performance analysis of verifiable data registry solutions for decentralized identifiers," in *2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2022, pp. 1–8. DOI: 10.1109/CSDE56538.2022.10089278.

[27] M. B. Jones, J. Bradley, and N. Sakimura, *JSON Web Token (JWT)*, RFC 7519, May 2015. DOI: 10.17487/RFC7519. [Online]. Available: https://www.rfc-editor.org/info/rfc7519 (Retrieved: 8/2025).

[28] K. N. Chadwick and J. Vercammen, "Openid for verifiable credentials," *OpenID Foundation*, 2022.

[29] E. Commission, "European blockchain services infrastructure (ebsi) vc framework," European Commission, Regulation, 2024. [Online]. Available: https://ec.europa.eu/digital-building-blocks/sites/display/EBSI/EBSI+Verifiable+Credentials (Retrieved: 8/2025).

[30] E. Commission, "European digital identity regulation," European Commission, Regulation, 2024. [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/eudi-regulation (Retrieved: 8/2025).

[31] I. D. S. Association, "International data spaces association," IDSA, Webpage, 2024. [Online]. Available: https://internationaldataspaces.org/ (Retrieved: 8/2025).

[32] GAIA-X, Webpage. [Online]. Available: https://gaia-x.eu/ (Retrieved: 8/2025).

[33] TANGO, Webpage. [Online]. Available: https://tango-project.eu/ (Retrieved: 8/2025).

[34] O. Terbu, T. Lodderstedt, K. Yasuda, D. Fett, and J. Heenan, "Openid for verifiable presentations 1.0," *OpenID Foundation*, Jul. 2025.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Near-Real-Time Fraud Detection in Banking Using HTAP Architecture: A Benchmarking-Driven Evaluation

Matteo G. Giorgino
Department of Computer Information Systems
Faculty of ICT, University of Malta
Msida, Malta
e-mail: matteo.giorgino.18@um.edu.mt

Joseph G. Vella
Department of Computer Information Systems
Faculty of ICT, University of Malta
Msida, Malta
e-mail: joseph.g.vella@um.edu.mt

***Abstract*—The increasing sophistication of financial fraud in the banking sector necessitates detection systems capable of operating on fresh transactional data and producing near-real-time analytical responses. Conventional architectures that decouple Online Transactional Processing (OLTP) and Online Analytical Processing (OLAP) via Extract-Transform-Load (ETL) pipelines struggle to meet these demands, particularly when strong data consistency and low-latency insights are simultaneously required. This paper investigates the suitability of Hybrid Transactional/Analytical Processing (HTAP) architectures for near-real-time fraud detection by consolidating transactional and analytical workloads within a single system. The evaluation is conducted using HyBench, a benchmarking framework designed to measure data freshness in centralized HTAP environments, extended in this work through a custom Python-based benchmarking harness. The harness orchestrates parameterised benchmark executions, enables systematic exploration of workload scenarios, and provides fine-grained performance monitoring beyond HyBench's native capabilities. Crucially, the HyBench schema employed in this study is modelled on a realistic banking transaction environment, capturing core entities and access patterns relevant to fraud detection, including high-frequency transactional updates and concurrent analytical queries. Over a 48-hour evaluation period, 96 parameterised benchmark runs are executed across multiple data volumes, with database configurations tuned to the available hardware. Results demonstrate that an HTAP platform can maintain continuous access to fresh transactional data, achieving sub-20 ms data freshness under mixed OLTP and OLAP workloads while sustaining high transactional throughput. Although performance trade-offs are observed when compared to standalone OLTP or OLAP systems, the findings highlight the importance of careful system configuration and tuning in balancing throughput, latency, and freshness. Finally, the extensible Python harness developed, enables the definition of custom metrics and integration of additional processing logic, offering a reusable foundation for realistic, fraud-oriented HTAP benchmarking beyond the scope of the original HyBench framework.**

Keywords–HTAP System; Near-Real-Time Fraud Detection; Database Architecture; Data Freshness; DBMS Benchmarking.

I. INTRODUCTION

Hybrid Transactional/Analytical Processing (HTAP) does offer timely and quality data for effective Fraud Detection [1], but successful deployment hinges on the availability of relevant, timely and high-quality data from integrating online and analytical systems present in financial institutions [2]. By scrutinising transaction records for distinctive patterns and anomalies, modern systems can distinguish legitimate activities from suspicious or malicious behaviour. The efficacy of these systems depends on computational capabilities that guarantee near-real-time data freshness, enabling rapid anomaly detection, and swift pattern recognition across vast datasets. The ultimate objective is to classify each transaction accurately and immediately, preventing fraud without impeding business operations.

In practice, Fraud Detection platforms must ingest data from a variety of heterogeneous sources. Each source contributes unique attributes that, when integrated, offer a comprehensive view of user behaviour. However, differences in data formats, transmission frequencies and endpoint capabilities can introduce synchronisation challenges. Data arrival is often asynchronous, leading to temporary mismatches between sources that must be reconciled to preserve analytical accuracy. Ensuring consistent data quality and low-latency access under such dynamic conditions requires robust strategies for data ingestion, harmonisation and error handling. Without adaptive synchronisation mechanisms, systems risk suffering latency spikes, incomplete data views or compliance gaps that undermine detection performance.

Many organisations have adopted architectures that separate Online Transactional Processing (OLTP) from Online Analytical Processing (OLAP), which are linked by Extract-Transform-Load (ETL) workflows. While this paradigm maintains a clear boundary between transactional consistency and analytical throughput, it introduces inherent latency and operational overheads. Batch-oriented ETL processes can struggle to satisfy the stringent low-latency requirements of fraud detection, and prevents OLTP systems from directly leveraging the most recent analytical insights during transaction authorisation.

This paper explores HTAP systems as a unified architecture capable of addressing these limitations. HTAP platforms merge transactional and analytical workloads, offering continuous, low-latency access to up-to-date data without the need for discrete ETL cycles [13-14]. By running analytics directly on live transactional feeds, HTAP

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

can enhance both responsiveness and detection accuracy, reducing the window of vulnerability to fraud. We present a detailed evaluation of HTAP performance in a banking context, measuring data freshness, throughput and anomaly-detection latency under realistic workload scenarios. To evaluate HTAP's effectiveness for near-real-time Fraud Detection, we select a benchmarking suite that defines and measures freshness on a centralised HTAP system, and extends it with a custom-built external harness. This setup allows for systematic scenario enumeration and detailed performance tracking across realistic banking workloads. Our findings demonstrate that, with careful configuration and tuning, HTAP systems can sustain sub-20 ms data freshness while processing high transaction volumes, outperforming OLTP/OLAP deployments in near-real-time fraud detection.

Despite their advantages, HTAP systems also come with limitations. Running transactional and analytical workloads concurrently can cause resource contention, particularly regarding memory management between differing storage formats. Yang et al. [18] characterize this conflict as a "Twisted Twin" phenomenon, where the memory demands of in-memory column stores (required for analytical speed) compete directly with row stores (required for transactional stability), potentially violating Service-Level Agreements (SLAs) critical to banking operations.

Tuning performance requires expertise, and some platforms may lack support for strong consistency or scalability. Upgrading from traditional architectures to HTAP based also involves significant cost and complexity.

This paper begins with Section I, which introduces the problem and motivation. Section II sets out the research aims and objectives. Section III provides background on fraud detection in banking, system development, the role of DBMS, database architectures, HTAP technologies, and benchmarking. Section IV details the schema, benchmarking suite, and experimental setup. Section V presents the results, focusing on the main KPIs. Finally, Section VI summarises the findings and outlines directions for future work.

II. Aims and objectives

The Fraud Detection System (FDS) aims to enable near-real-time classification of fraudulent transactions by ensuring data freshness, and facilitating ad hoc pattern recognition, thereby addressing the operational limitations of traditional FDS. In achieving this aim, a set of objectives was laid down:

1) Conducting a literature review to explore available HTAP configurations that support both a transactional banking system and an analytical engine for data analytics.
2) Choosing and adapting an HTAP benchmarking suite, including setting up both separate OLTP and OLAP components for testing and evaluation on a modern and open source DBMS.
3) Developing an external orchestration harness that coordinates the complete benchmarking pipeline in a plug-and-play manner, without intrusive changes to the benchmark or DBMS source code.
4) Running operations on the OLTP system and executing transaction classification requests on the OLAP system in both the HTAP and two-system setups. This requires using a benchmarking suite to measure key performance metrics, which runs over a wide array of HTAP and DBMS configurations.
5) Evaluating the performance of various runs using the chosen HTAP configuration and different data movement scenarios.

III. Literature review

A. Fraud Detection in Banking

Fraud is defined as any activity that relies on deception to achieve a gain. Banks are among the most obvious targets for fraudsters seeking financial gain. Given the complexity of modern fraud schemes and the sophistication of fraudulent behaviour, the financial industry faces some of the toughest detection and prevention challenges, necessitating advanced systems that monitor multiple sources of data and activities.

A recent study revealed that fraud in the banking industry has become a matter of grave and global concern, causing significant financial and non‑financial damages to banks, customers, other stakeholders and economies [3]. Fraud in banking also results in reputational damage and compliance challenges. Regulatory compliance further necessitates robust Fraud Detection frameworks to protect sensitive data and ensure adherence to legal practices.

B. Fraud Detection Systems Development

Traditional FDS primarily rely on rule‑based approaches, which flag suspicious activities based on predefined criteria, such as unusually high transaction amounts, frequent transactions, or geographical discrepancies [4]. These static rules struggle to adapt to the continuously evolving tactics employed by fraudsters.

Machine Learning (ML) techniques are increasingly being integrated into traditional FDS to address some of these limitations, as discussed by Minastireanu [5]. These models typically employ supervised learning models trained on historical data [6]. They exhibit reduced efficacy against novel fraud schemes that deviate significantly from historical fraudulent patterns. Additionally, these ML approaches often require substantial time and resources for periodic retraining and validation to maintain their relevance.

Fraud Detection systems also make use of database-bound workloads to execute processes. These refer to the pattern of operations a database must handle, defined by the sequence and ratio of read and write queries, along with their transactional consistency requirements. These workloads are not static; they evolve continuously based on end-user behavior. As user interactions are often irregular and difficult to predict, the resulting workload can fluctuate significantly over time, placing varying demands on the database system.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A further notable limitation of traditional systems is the separation of the transactional data, which is used in day-to-day activities, from the analytical data utilised in reporting and analysis. These distinct operational environments are often optimised differently, as discussed by Camilleri [17], impeding the integration of near-real-time transaction analysis with historical data evaluation. Lastly, the broader advancement of Fraud Detection methodologies suffers due to restricted knowledge-sharing within the public domain.

C. Database Management Systems for Fraud Detection

Database Management Systems (DBMSs) are critical components in Fraud Detection architectures, serving as the backbone for processing large volumes of transactional and analytical data. As banks grapple with increasing transaction volumes and sophisticated fraud schemes, traditional DBMSs face limitations, prompting the exploration of more advanced DBMS architectures that can provide near-real-time responses, robust scalability, and comprehensive data integration capabilities.

A DBMS ensures that data remains available to users and applications, handles increasing volumes of data without loss of data consistency, and is able to scale to meet rising demand. DBMSs play an increasingly crucial role in supporting Fraud Detection mechanisms in banking. The ability of a DBMS to process, store, and retrieve data efficiently in a structured and unstructured manner can significantly improve the quality of Fraud Detection efforts.

At its core, an FDS must rapidly ingest large volumes of heterogeneous data, enforce stringent data quality and integrity constraints, and support ad hoc analytical queries. Nevertheless, Relational DBMSs have encountered bottlenecks in performance and scalability, especially in transactional processing [7]. Detecting patterns of suspicious transactions requires the ability to query and analyse historical data alongside online transactions.

D. OLTP and OLAP Workloads in Banking Systems

Each workload type imposes distinct and often conflicting demands on DBMSs, making their coexistence a significant architectural challenge.

OLTP workloads are responsible for handling high-frequency, low-latency transactions generated by day-to-day banking operations, such as card payments, transfers, and account updates [19]. These workloads are typically write-intensive, involve simple queries operating on small data subsets, and require strict consistency and durability guarantees to ensure data correctness. In the context of fraud detection, OLTP systems must ingest transactions reliably and at scale, often under stringent latency constraints, as delayed processing can directly impact fraud prevention effectiveness.

OLTP workload levels naturally vary over time, shifting between periods of high and low demand depending on user activity. In addition to volume changes, the scope of the workload can also fluctuate, with some periods experiencing minimal data contention and others facing significantly higher contention. When many transactions compete for the same data, this increased contention can lead to reduced throughput and slower overall system performance.

On the other hand, OLAP workloads focus on complex, read-intensive analytical queries over large volumes of historical data. These queries commonly involve aggregations, joins, and statistical computations aimed at identifying long-term patterns, anomalies, or behavioural trends. For fraud detection, OLAP workloads enable retrospective analysis, model training, and the evaluation of historical transaction behaviour to uncover subtle or evolving fraud strategies. Performance in OLAP systems is optimised for throughput and query complexity rather than transactional latency [19].

Traditionally, OLTP and OLAP workloads have been supported by separate systems, with transactional data periodically transferred from operational databases to analytical platforms via Extract, Transform, and Load (ETL) pipelines. While this separation allows each system to be individually optimised, it introduces significant latency, data staleness, and operational complexity. In HTAP systems we also find OLxP workloads, which combine transactional and operational operations, enabling the hybrid execution of near-real-time processing alongside concurrent analytical queries.

E. Limitations of ETL for Fraud Detection

This batch-oriented nature of traditional ETL architectures, combined with the long lead time for each ETL cycle, results in significant delays between when a transaction occurs and when it is available for analysis [7]. These delays make traditional ETL poorly suited for near real-time use cases, such as Fraud Detection, where up-to-date information is crucial for identifying and preventing fraudulent activities. Moreover, traditional ETL systems are typically monolithic and rigid, making them difficult to scale and adapt to new data sources.

Modern approaches attempt to address the contrived nature of ETL processes through various tactics, such as:

i. Sophisticated ETLs incorporate dependence management, parallel processing, data dictionaries, and robust software engineering practices [7], yet these still fall short of meeting all OLAP requirements, like data freshness.

ii. Techniques that aim to reduce the ETL processes by directly connecting data sources to data cubes [19]. Although this reduces the transformation process, it often results in compromised data quality and reduced flexibility for accommodating new analytical requirements.

iii. Streaming data from disparate sources is another approach [17], but this introduces additional levels of ETL to manage data quality issues, and the streaming channel semantics.

In the context of Fraud Detection, these challenges render traditional ETL architectures less effective, as they struggle to meet the demands of near-real-time processing, data freshness, and continuous data ingestion.

To address these challenges, the DBMS landscape has diversified into multiple directions. Noticeably, columnar with in-memory analytical platforms, which integrate OLTP and OLAP workloads within a single engine, enabling

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

HTAP. By maintaining online materialised views or employing Multi-Version Concurrency Control (MVCC) optimised for OLxP mixed workloads, HTAP systems permit on-the-fly computation of fraud scores without the latency penalties of ETL pipelines [8].

Practical performance depends heavily on real-world data patterns, which are driven not just by the underlying schema but by unpredictable end-user behaviour, which make static optimisation insufficient. This unpredictability creates additional challenges in scaling transactional throughput while preserving data consistency guarantees.

F. Database Architecture for Fraud Detection

In many architectures, OLTP and OLAP systems operate and set-up in isolation. OLTP databases are optimised for light transactions with few tuples in its scope, handling inserts, updates and deletes with ACID (Atomicity, Consistency, Isolation and Duration) guarantees, whereas OLAP engines are tuned for complex, read-heavy queries against large datasets. To bridge these worlds, ETL pipelines are constructed to periodically move data from OLTP systems into purpose-built Data Warehouses (DWH) to support OLAP. This also introduces latency: data freshness is bounded by the ETL cadence, and the overhead of maintaining dual schemas with the pipeline code can be substantial [9].

There are two brazen challenges organisations face to maintain a data infrastructure that caters for their data driven decision making, namely; (i) performance in terms of query response time, transactional throughput, and resilience to computational failures, and (ii) moving data from internal and external data sources to build an integrated, synchronised, business-process focused, and time-variant data repository, i.e., a Data Warehouse [10].

If data freshness becomes a requirement, as in a banking FDS, then the architecture with ETL does not meet it [9]. HTAP architectures seek to collapse the boundary between OLTP and OLAP by introducing a computational set-up that supports both workloads within a single, integrated engine and instance, i.e., OLxP modality.

This unified approach can dramatically reduce data movement overheads, eliminate ETL-induced data staleness, and simplify system maintenance. However, it does demand advanced engine optimisations and comes at a loss of the decoupling and staging flexibility offered by ETL pipelines.

G. HTAP Architectures and Techniques

HTAP, introduced by Gartner in 2014 [13], describes a database architecture that unifies transaction processing and analytics in near-real-time, allowing OLTP and OLAP workloads to run side by side without undue interference, as seen in Figure 1. By collapsing separate systems and eliminating complex ETL pipelines [12], HTAP simplifies data management. However, supporting both transactional and analytical demands simultaneously remains challenging, given their inherently different performance and resource requirements. To address these performance isolation challenges in mission-critical environments, adaptive mechanisms such as Bayesian optimization frameworks have been proposed to dynamically reallocate memory and optimize data synchronization strategies based on fluctuating workload patterns [14].

Overall, supporting both transactional and analytical queries on the same dataset increases the risk of data contention in HTAP systems, necessitates careful management of isolation levels, concurrency controls, and workload balancing strategies is essential.

HTAP remains a conceptual model rather than a formally standardised architecture, and there is currently no broad consensus on the best way to implement it.

In their HTAP Systems Survey, Zhang et al. [9] neatly summarise the main HTAP storage architectures, each designed to balance the demands of the transactional and analytical workloads. One common architecture is the primary row storage with an in-memory column store, where the primary storage is a row-oriented database optimised for transactional operations, and an in-memory column store is used to handle analytical queries. This design facilitates real-time analytics without compromising transactional performance. Another approach is the distributed row store with a column store replica, where a distributed row-based storage system manages transactional data, while a replicated columnar store is used for analytics. This replication ensures that analytical queries do not interfere with transactional operations, maintaining system efficiency. A variant of this design is the primary row store with a distributed in-memory column store which utilises a row-oriented primary storage for OLTP workloads and a distributed in-memory column store for OLAP tasks. The distributed nature of the column store allows for scalable and efficient analytical processing.

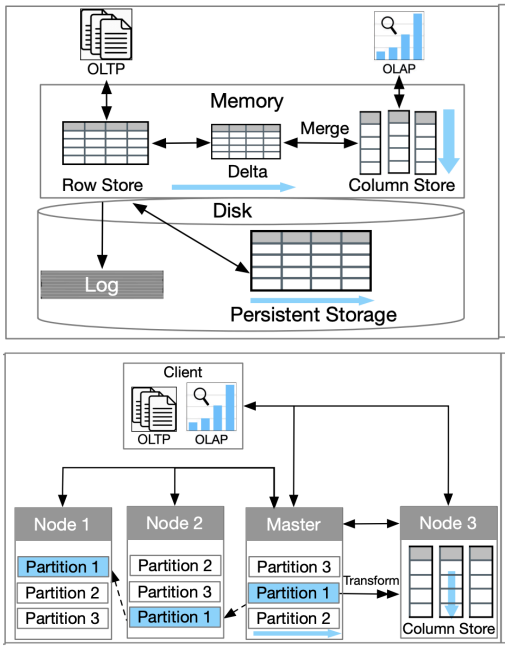


Figure 1. HTAP Architectures with Primary Row Store + In-Memory Column Store and Distributed Row Store + Column Store Replica [9]

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

The choice of architecture depends on specific application requirements, workload characteristics and resource availability. For instance, high-volume transactional systems favour OLTP for fast, consistent writes, while analytics-heavy workloads benefit from OLAP's read-optimized design. Hybrid systems like HTAP are more suitable when near-real-time insights are needed without sacrificing transactional integrity, such as in fraud detection scenarios. Additionally, resource constraints (e.g., limited memory or compute power) may dictate the use of simpler architectures, whereas larger organizations with more capacity can adopt in-memory or distributed architectures for greater responsiveness and scale.

The primary challenge in an HTAP system is to facilitate the smooth flow of data from all these different sources into a unified platform where both transactional and analytical operations can occur, as shown in Figure 2. Ensuring that transactional updates and analytical views remain tightly consistent is critical for applications that demand ACID guarantees, such as Fraud Detection in banking. Unlike eventual-consistency transactional models, where updates propagate to distributed DBMSs asynchronously and with no agreement, and stale reads are tolerated until "eventual" convergence, banking environments cannot risk inconsistencies in account balances or transaction histories. Thus, HTAP platforms in banking must implement strong consistency mechanisms to synchronise updates from OLTP to OLAP stores with minimal latency and without sacrificing throughput.

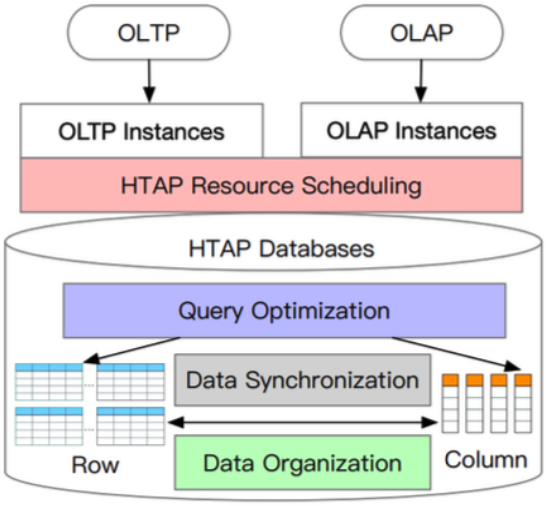


Figure 2. HTAP System Overview [16]

HTAP surveys [16] identify two principal synchronisation techniques. The first is in-memory delta merging which periodically integrates recent transactional writes, held in a small, write optimised delta store, into the main columnar analytical store based on a threshold such as elapsed time. This approach keeps ingestion paths fast while bounding the merge overhead, but can introduce micro-latency windows of unsynchronised data. Secondly, log-based delta merging, which by contrast, streams redo logs or change-data-capture events directly into the analytic engine via multi-level merge pipelines. Though more complex, these pipelines can achieve sub-second freshness and preserve strict isolation by relaying only committed transactions.

H. Data Ingress

Data ingress refers to how data is received and incorporated into a system. Banking systems typically generate vast amounts of data from multiple sources. The primary challenge in an HTAP system is to facilitate the smooth flow of data from all of these different sources into a unified platform where both transactional and analytical operations can occur.

In some database systems, data ingress is typically handled in batches. For instance, transactional data might be processed in large chunks at periodic intervals [7]. However, this approach would delay Fraud Detection because the analysis occurs after the transaction is completed. HTAP systems, on the other hand, can ingest data continuously, ensuring that Fraud Detection algorithms have access to the most up-to-date information in near real-time. By integrating multiple data streams, HTAP systems can rapidly identify suspicious activity, reducing the window of opportunity for fraudulent transactions.

Modern HTAP engines achieve this through a DBMSs multi-model storage, combining row-stores for low-latency writes, document stores for unstructured data, and full-text or vector indexes for unstructured content, while a single query optimiser pushes aggregations down into the appropriate layer [9]. This integration allows for more comprehensive Fraud Detection algorithms that can analyse data from various sources and identify patterns that might indicate fraud.

I. Benchmarking

Benchmarking database systems underpins objective performance assessment, offering repeatable tests that measure metrics, such as throughput, latency or completion time against standardised workloads [12]. A typical benchmark defines a data schema, data volume, workloads (queries, transactions, frequency, scope and sequence) and an evaluation criteria. These tests serve multiple purposes: identifying raw horsepower through intensive tuning; comparing enhancements or configuration changes; and contrasting architectures or hardware choices under controlled conditions. Fairness, consistency and conservatism in benchmark design are essential to yield meaningful insights [22].

Traditional OLTP benchmarks such as TPC-C, an industry standard since 1992, with the latest updates in 2010, simulate transactional workload with a mix of order-entry, payment, stock-level, delivery and order-status transactions, measuring performance in transactions per minute (tpmC) under stringent ACID constraints [23]. Complementing this, decision-support benchmarks like TPC-H provide a suite of ad hoc, complex SQL queries over a star-schema, designed to stress-test query optimisation, I/O bandwidth and large-scale data processing

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

typical of OLAP environments [23]. Together, these benchmarks have long guided practitioners in sizing hardware, tuning configurations, and comparing system capabilities for purely transactional and analytical workloads.

HTAP systems, which handle both OLTP and OLAP workloads, require specialised benchmarking frameworks to capture their unique characteristics. Traditional benchmarks such as TPC‑C and TPC‑H focus on OLTP and OLAP workloads respectively, making them insufficient for evaluating HTAP systems [9]. This gap has led to the development of HTAP-specific benchmarks.

Early efforts, such as CH-benCHmark extend TPC-C and TPC-H by executing transactional and analytical workloads on the same dataset, thereby revealing workload contention effects [15]. However, CH‑benCHmark does not fully capture real‑world HTAP workload complexities, as it does not integrate mixed workloads in a way that reflects modern HTAP applications.

HTAPBench [20] was later introduced to address some of these limitations by providing a more representative hybrid workload. It includes both transactional and analytical queries while ensuring that the benchmark covers workload interference, data synchronisation, and real‑time query performance. However, it lacks a strong focus on anomaly detection and time-dependent workload patterns, which are crucial for financial and operational analytics.

HATtrick [21] is a recent HTAP benchmark that introduces a systematic approach to evaluating HTAP systems by focusing on two key aspects: performance isolation and data freshness. Performance isolation ensures that OLTP and OLAP workloads do not interfere with each other, maintaining system efficiency under mixed‑mode workloads. HATtrick also introduces the concept of the throughput frontier, a performance metric that maps the maximum achievable transactional and analytical throughputs across various workload mixes. By integrating these concepts, HATtrick provides a robust method for evaluating HTAP systems' ability to handle hybrid workloads efficiently while maintaining data consistency and minimal latency.

HyBench [16] is another significant HTAP benchmark designed to address gaps left by previous benchmarking frameworks. It integrates realistic banking transactions alongside analytical tasks like trend analysis. Its workload comprises 18 transactional operations and 13 analytical queries executed under mixed scenarios, mirroring real-world HTAP use cases more closely than its predecessors. By focusing on financial workloads and anomaly-driven scenarios, HyBench offers a comprehensive evaluation framework that blends transactional throughput, analytical query performance and data freshness. Its extensible design allows practitioners to adjust parameters, incorporate new metrics or inject domain-specific logic, making it a versatile tool for advancing HTAP research and guiding production deployments in latency-sensitive environments.

IV. Design, Implementation and Testing

A. System Design

HyBench forms the foundation of our evaluation framework, simulating both transactional and analytical workloads on a unified schema mimicking typical banking entities, as seen in Figure 3, drawing inspiration from industry leaders such as Alipay and WeBank [16]. The necessity for real‑time decision‑making is exemplified by Alipay's 3‑1‑0 model, where a loan application is submitted in three minutes, assessed in one second, and approved without human intervention. Similarly, WeBank employs HTAP capabilities to automate loan approvals by identifying credit fraud risks within a minute.

HyBench's system design revolves around several core components: a synthetic data generator, a workload executor, and benchmarking metrics. The data generator is crucial in creating a realistic yet synthetic data environment, mimicking a typical banking schema without exposing any sensitive data. By ensuring a balance between transactional integrity and analytical accuracy, the design aligns well with the requirements of Fraud Detection in banking.

The schema incorporates key features for risk management: the DELINQUENCY field flags overdue loans, while the ISBLOCKED field restricts fraudulent users and accounts. Additionally, the schema differentiates between synthetic timestamps (TIMESTAMP) and real‑time timestamps (FRESH_TS) to evaluate data freshness in analytical queries, referred to as F-Score. This metric quantifies the delay between a transactional update and its visibility to analytical queries.

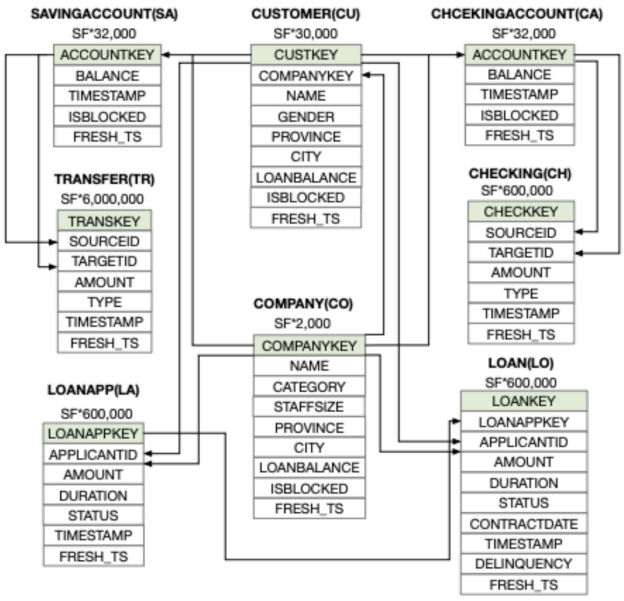


Figure 3. HyBench Schema with Volume Indicators [16]

For a query Q, with a result set in OLTP and OLAP denotes as R_{TP}^{Q} and R_{AP}^{Q} respectively; each tuple in the result set is defined as $r = (i, d, ts)$, where i is the tuple ID, d

is the data and *ts* is the timestamp. The F-Score is then defined as follows by the authors of HyBench:

$$\overline{f_s} = \max \begin{cases} ts_{r_a} - ts_{r_b} \mid r_a \in R^Q_{TP}, r_b \in R^Q_{AP}, i_{r_a} = i_{r_b} & \text{update} \\ ts_Q - ts_{r_a} \mid r_a \in R^Q_{TP}, r_b \notin R^Q_{AP}, i_{r_a} = i_{r_b} & \text{insert} \\ ts_Q - ts_{r_a} \mid r_a \notin R^Q_{TP}, r_b \in R^Q_{AP}, i_{r_a} = i_{r_b} & \text{delete} \end{cases} \quad (1)$$

where max takes the maximum value of the timestamp differences between the result sets of R^Q_{TP} and R^Q_{AP} concerning the cases of the update, insert and delete. t_{SQ} is the starting time of the query *Q*. The F-Score is calculated during the execution of the hybrid workloads, specifically when both transactional and analytical workloads access the same data in parallel. The TP client first executes a read operation and logs the returned timestamp. Immediately after, the analytical client runs the query and also logs the timestamp, then formula (1) is applied to extract a value for freshness.

The Customer table (CU) includes attributes like customer ID, demographics e.g., name, gender, province, city, loan balances, and an ISBLOCKED flag, enabling customer segmentation and detection of location‑based fraud patterns. The Account tables (i.e., SavingAccount (SA) and CheckingAccount (CA)) detail account balances, timestamps, and blocked statuses, facilitating the identification of unusual account behaviours, such as multiple blocked accounts tied to a single customer. Transactional data captured in the Transfer table (TR), including transaction amounts, types, and timestamps, allows for anomaly detection in transaction frequency.

This structure ensures that HyBench effectively simulates the multi-nature of HTAP workloads, where transactional efficiency and analytical accuracy are equally critical in real-world financial applications.This schema is also scalable, with scale factors (SF) applied to tables to simulate varying system loads, where a SF of 1x corresponds to approximately 1GB, 10x to approximately 10GB and 100x to approximately 100GB.

Inspired by the HATtrick [21] benchmark, HyBench models real-world financial transactions with proportional relationships between tables [16]. For example, the Customer, Company and Transfer tables follow a similar scaling to the corresponding entities in previous benchmarks, while the Loan Applications, Loan and Checking tables have reduced sizes to reflect lower frequency of loan applications and cheque transactions compared to everyday bank transfers.

Additionally, geographical correlation is introduced, meaning customers and businesses within the same city are more likely to engage in financial interactions. The dataset is generated in three phases, aligned with real‑world economic trends: the years (1) 2014‑2019 focus on foundational entities such as Customer and Company accounts, (2) 2019‑2023 models standard financial transactions like Transfers and Cheques, and (3) 2023‑2024 introduces Loan Applications and Approvals, reflecting the increasing prevalence of digital lending.

Beyond realistic data generation, HyBench overlaps anomaly generation to support Fraud Detection and risk control use cases. Anomalies such as blocked accounts and delinquent loans are injected based on probability thresholds, ensuring that fraudulent behaviours appear naturally within the dataset. These anomalies are generated using a query‑driven approach, where analytical queries pre‑identify risk patterns, and transactions satisfying those conditions are recorded for anomaly creation. A sampling technique is used to maintain a curated set of anomalies efficiently, preventing performance overhead while ensuring diverse fraud scenarios. This method enables HyBench to evaluate HTAP systems' ability to detect and respond to fraudulent activities in near real‑time, making it a valuable tool for benchmarking Fraud Detection mechanisms in financial applications. Notably, many of these features can be configured at run time, allowing flexible experimentation under different workload patterns and system conditions.

A harness orchestrates the workflow: it restores a clean database snapshot, vacuums and analyses tables to eliminate fragmentation, pre-warms the instance cache using a representative query, then launches concurrent clients for:

- Transactional Processing (TP) workloads, with high-volume, short-lived OLTP operations
- Analytical Processing (AP) workloads, with long-running, scan-intensive OLAP-style queries
- Hybrid Processing (XP) mixed workloads, combining both TP and AP workloads

All runs were conducted on a standalone machine (macOS, M3 Pro, 16 GB RAM, 500 GB SSD), using PostgreSQL 16, with pg_stat_statements enabled to record detailed query metrics. The Python 3.13.2 harness performed orchestration and monitoring, while Java components (via OpenJDK 21) drove the HyBench workload. Key PostgreSQL configurations were tuned, allowing up to 100 concurrent connections, allocating 4 GB each to shared_buffers and effective_cache_size, disabling autovacuum, and setting work_mem to 64 MB. These configurations were chosen based on the available hardware capacity and industry best practices.

During each benchmark, we gathered metrics from both HyBench logs and the Python harness, as seen in Table I.

TABLE I. DATABASE METRICS COLLECTED DURING BENCHMARKING

Category	Metrics Collected
Query Performance	Total Execution Time, Mean Execution Time, Number of Calls, Number of Rows Returned, Shared Block Hits, Shared Block Reads, Query Text.
Table Activity	Number of Inserts, Updates and Deletes.
Lock Information	Lock Type and Number of Locks Held.
Transactional Throughput	Number of Committed and Number of Catered-for Transactions per Table.
Storage Statistics	Schema Name, Table Name, Total Table Size and Total Index size.

Together, these measurements offer a comprehensive view on resource use and contention under concurrent

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

OLTP/OLAP conditions. To unify these dimensions, HyBench's creators introduced the H-Score. This unified metric incorporates throughputs, data freshness seen in (1) and scale factor to yield an overall performance rating.

The throughput KPIs considered by H-Score are the following: Transactions per second (TPS) which measure how many complete transactional operations a system can process in one second, Queries per second (QPS) capture the rate at which read-only queries are executed, and for the mixed workloads, both transactional (XP-TPS) and analytical (XP-QPS) operations run concurrently, and their combined throughput is expressed as XPS, representing the overall processing capacity of the system.

The H-Score is defined as the geometric mean of all throughputs, multiplied by the SF and divided by the freshness metric, as seen in (2).

$$H\,Score = SF \times \frac{\sqrt[3]{TPS \times QPS \times XPS}}{f_s + 1} \quad (2)$$

H-Score is beneficial as solely relying on one aspect cannot reflect the true HTAP performance. The five components in H-Score are widely recognised by benchmarking suites as the most important factors for quantifying the HTAP performance [11].

B. Implementation

We defined 11 core parameter dimensions to explore various workload mixes and data volumes, producing 144 unique configurations, as seen in Table 1, but we ultimately retained only 96 combinations (i.e., the 1x and 10x SF ones) after observing inconsistent KPI behaviour at 100x, namely scalability limits in HyBench's threading and PostgreSQL's I/O performance on given computer set-up.

By adjusting these parameters, we could evaluate scenarios with standard transactional-heavy loads (3:1 TP:AP) and more analytics-intensive mixes (3:2 TP:AP), as well as stress-test at 100x SF.

One of the key components developed for this project is the Python harness. At a high-level, the harness is responsible for generating the necessary properties files automatically, executing the HyBench benchmark, performing database housekeeping tasks and running queries in parallel to extract the required metrics. The harness script handled the below process:

1) Generates the HyBench '.props' files programmatically for every parameter combination.
2) Restore and reset the database from a known backup via 'pg_restore' utility, then execute VACUUM FULL ANALYZE on all tables to rebuild storage and refresh data dictionary (DD) statistics.
3) Prewarm the cache using a realistic query that joins key tables and exercises index and sequential scans in parallel, ensuring relevant pages are memory-resident.
4) Launch HyBench with a '.props' configuration file, while a background process queries the DD views, pg_stat_statements, pg_locks, pg_stat_user_tables and pg_stat_database every 60s for live metrics.
5) Parse the logs and plot key performance indicators.

Initially, prewarming full tables was tested but quickly abandoned due to significant memory constraints at larger scale factors. Prewarming only indices was then explored, but the performance gains were negligible compared to the cold-cache run. Ultimately, prewarming the database using a representative query was selected as the best approach, as it realistically populated the cache by involving table joins, filters and aggregations that mirror typical HyBench workloads.

TABLE II. HYBENCH PARAMETER CHOICES

Parameter	Set Value	Description
sf	1x, 10x, 100x	Scale factors for the table data
at_percentages	(35,25,15,15,7,3), (3,7,15,15,25,35), (10,10,20,20,20,20)	AT Ratio (sum = 100%)
apclient	10	AP concurrency
tpclient	15, 30	TP concurrency
fresh_interval	150	Freshness evaluation is done every (xpRunMins/150) s
apRunMins	5, 10	AP evaluation time
tpRunMins	5, 10	TP evaluation time
xpRunMins	5, 10	XP evaluation time
xapclient	10	XP-ATS concurrency
xtpclient	15, 30	XP-IQS concurrency

The query execution plan provided important insights into its suitability for prewarming. The query spans several large tables, namely Company, Transfer, Customer, SavingAccount, CheckingAccount, and LoanApps, and involves various operations such as grouping, sorting, and joining. Notably, the query makes extensive use of index scans, particularly on the company_pkey and customer_pkey indexes, which helps to minimise unnecessary disk I/O. The hash join between LoanApps and Transfer, as well as the nested loop joins on other tables, indicate that the query exercises key parts of the database's query planner and cache management systems. The query also utilises parallelism, with two workers launched to handle different parts of the data simultaneously. This parallel execution strategy aids in speeding up the data processing and in more efficiently filling the cache. The result (for the 10x SF) is a significant number of buffer hits and disk reads, ensuring that the most relevant data pages are loaded into memory before the benchmarking begins. The query also performs multiple index scans, which further assist in warming the cache by making use of existing indexes and reducing the need for full table scans. The query's execution plan also revealed that significant portions of the data were processed by the sequential scan

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

operations, especially on the LoanApp table, which contains over 2.4 million rows. This was expected, given that the filter on the type column in the transfer table is not highly selective, and was therefore removed from the query.

C. Testing

With the system design and implementation stages complete, we now move onto the testing phase, which builds on that foundation to evaluate whether the system performs effectively under real-world conditions.

The primary aim of this paper is to assess how well the HTAP approach performs compared to a more traditional OLTP-OLAP setup, specifically in scenarios where both transactional and analytical operations must occur concurrently.

An important part of the evaluation involves measuring and interpreting metrics derived from the runs over the implementation just described. These metrics help determine whether the system can respond quickly enough for it to be useful in Fraud Detection scenarios, where delays in insight generation can lead to financial risks.

A few metrics were identified, including query execution latency, throughput, and consistency across different workload mixes and database scales. Equally important is the observation of how the database instance manages available computational resources such as buffer memory, index sizes, and disk usage during these operations. This helps assess whether performance gains come at a cost of excessive storage and resource consumption. In addition to raw performance, testing aims to validate whether the system can handle a balanced stream of transactional reads/writes and analytical reads without one negatively impacting the other.

We began with unit tests of individual harness modules (e.g., DB housekeeping, monitoring, HyBench). Next, integration tests ran end-to-end workflows using varied '.props' files to validate the sequence: restore → housekeeping → prewarm → workload → monitoring. Logs from PostgreSQL and HyBench were cross-verified to ensure consistency, accounting for differences in granularity (e.g., DBMS internal vs. atomic operations launched by HyBench). All runs were performed in a controlled environment with server resource prioritisation to avoid external interference. Three main scenarios were executed with each taking 24 hours:

1) 1x: Establish expected performance and tune PostgreSQL parameters to avoid configuration‑induced artefacts.
2) 10x: Stress test concurrency, observe degradation in freshness and throughput, and validate that HTAP sustains desired freshness under mixed workloads.
3) 100x: Identify limits of the harness and the hardware set-up available for the DBMS, revealing thread‑management issues, JDBC termination overheads, and I/O saturation that rendered metrics unreliable (consequently it was discarded for evaluation).

For each run, we generated data (via HyBench's gendata module), created indexes and tables, and ran the full benchmark cycles.

In addition to the above testing, a series of validation checks were incorporated to ensure that the benchmark executions were functioning correctly and consistently across all runs. These checks focused on monitoring both the structural integrity of the database and the operational behaviour of the workload. Specifically, metrics such as table sizes, index sizes, and the volume of insert and update operations were captured and plotted following each full benchmark run. These metrics were collected from the pg_statements data dictionary. This allowed for a clear, visual confirmation that the system was maintaining stable growth patterns and executing operations as expected. Any significant deviation in these indicators would have signaled potential issues such as workload misconfigurations, storage anomalies, or transactional failures, none of which were observed during the monitored runs.

To further validate the reliability of the collected performance metrics, the key throughput indicators (i.e., TPS, QPS and XPS) were compared qualitatively against those reported in the original HyBench paper for PostgreSQL [9]. While absolute figures were expected to differ due to variations in hardware and environment, the relative performance trends and ratios aligned well with the reference benchmarks. We also compared the overall metrics obtained from our 1x, 10x, and 100x runs to ensure everything was working as expected. It was at this point that after executing the 100x run, the metrics produced did not make sense, so we decided to investigate further. The throughput metrics across all modes plummeted for the 100x run, falling below one transaction or query per second, which is an unsustainable level that would render banking systems at scale unusable. Correspondingly, the H-Score for the 100x run was exceedingly poor, confirming that these harness-based artifacts, together with the systems under test, were driving anomalous performance. Moreover, in a benchmarking context there is very little room to tweak or patch the harness source code on the fly as any such changes would undermine comparability, so the only viable recourse was to acknowledge these limitations and exclude the 100x results from further analysis. The 100x run results have consequently been discarded from the forthcoming evaluation.

V. Evaluation

At a scale factor of 1x, throughput measurements across all execution modes exhibit substantial variability, as shown in Figure 4. This behaviour is primarily driven by fixed system overheads such as query compilation, warm-up, cache population, and operating system scheduling, which dominate overall execution time when workloads are small. As a result, normalized throughput values are generally low and unstable (shown by the narrow box plot with a low mean), with several runs producing disproportionately high values, shown by the outliers, due to favourable transient conditions, such as warm caches or reduced background contention. Consequently, performance at this scale factor is

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

more reflective of execution overhead than that of the system's true throughput capabilities.

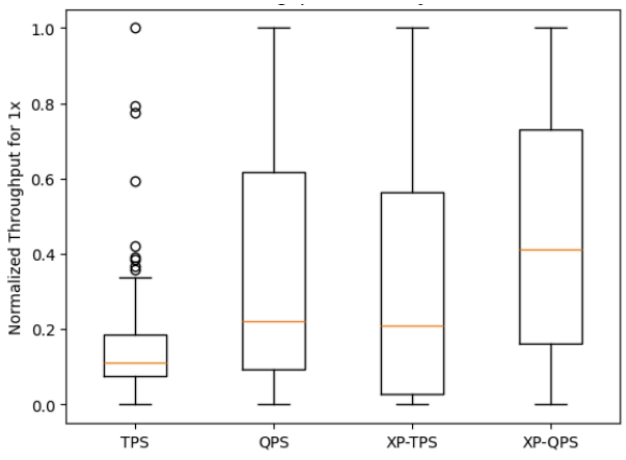


Figure 4. TPS vs QPS vs XP-TPS vs XP-QPS over all 1x Runs

When the workload is increased to a 10x scale factor, throughput measurements become significantly more stable and consistent, as seen in Figure 5. The larger workload amortizes fixed overheads, allowing the system to operate in a steady-state regime where resource utilization, such as CPU, memory, and I/O, becomes the primary performance determinant. This is reflected in higher median normalized throughput values and tighter interquartile ranges, particularly for baseline TPS and QPS metrics. The reduced presence of extreme outliers further indicates that system behaviour is less sensitive to transient effects, making the results at this scale factor more representative of real-world performance.

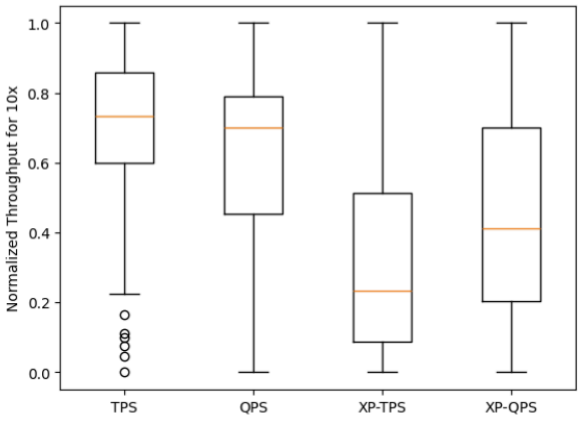


Figure 5. TPS vs QPS vs XP-TPS vs XP-QPS over all 10x Runs

Comparing execution modes, baseline TPS and QPS demonstrate strong scalability and convergence as the scale factor increases, indicating that these execution paths effectively exploit increased workload parallelism and maintain predictable performance under load. In contrast, mixed workload throughputs show greater variability, particularly for transactional workloads. XP-TPS exhibits a lower median throughput and wider dispersion at the 10x scale, suggesting increased sensitivity to contention, synchronization, or coordination overheads. XP-QPS, while also more variable than its baseline counterpart, benefits more from higher scale, likely due to batching and analytical optimizations that become more effective as workload size increases. These box plots also highlight the importance of scale when evaluating throughput performance in HTAP-oriented systems. Low-scale benchmarks may hide true system behaviour due to overhead-dominated execution, whereas higher-scale workloads provide more reliable insight into scalability, execution efficiency, and the trade-offs introduced during testing.

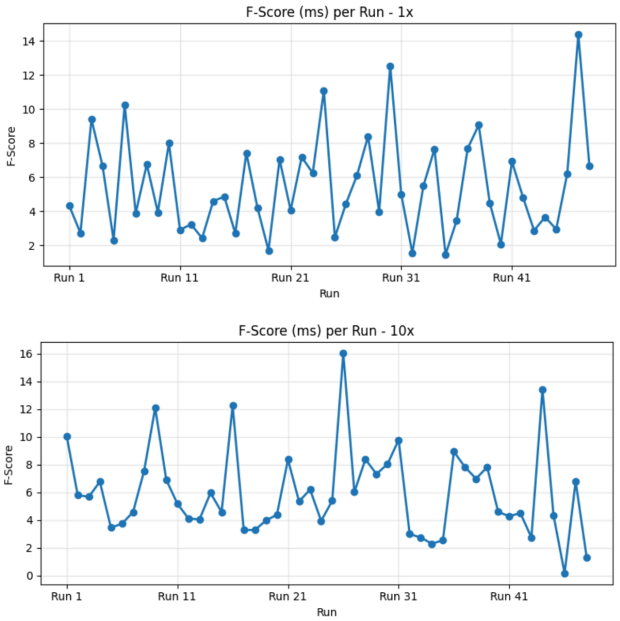


Figure 6. Freshness (F-Score) Comparison between 1x and 10x

The freshness obtained from the benchmarking run, represented by the F‑Score in Figure 6, provides insights into data latency experienced during analytical queries across the benchmark runs at both 1x and 10x scales. At the 1x scale, freshness typically fluctuates between 2 and 12 milliseconds, exhibiting frequent but manageable variations. Occasional peaks reaching upwards of 12ms indicate moments of increased resource contention, delaying the propagation of transactional data to analytical queries. At the 10x scale, freshness profiles are similar in overall behaviour, but exhibit slightly higher peak latencies, up to around 15ms. These higher peaks suggest intensified resource competition for certain parameter configurations, reflecting inherent challenges faced by the HTAP system in maintaining strict freshness under heavier concurrent workloads. Nevertheless, these freshness intervals remain within acceptable thresholds for most near real‑time use‑cases, as they demonstrate the system's ability to consistently provide timely analytic visibility.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

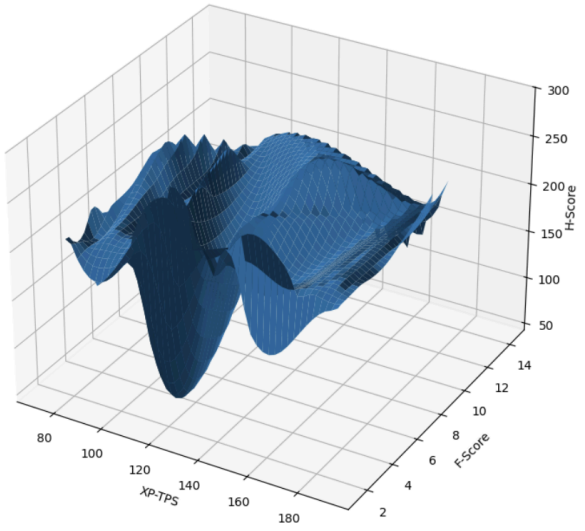


Figure 7. Impact of XP-TPS and Freshness on H-Score in 1x Runs

The surface plot shown in Figure 7 illustrates how the H-Score varies with respect to XP-TPS and freshness during the 1x scale runs; however, it is important to note that this relationship is conditional rather than exhaustive. While XP-TPS and freshness are explicitly represented on the axes, the H-Score is a composite metric that is also influenced by other performance dimensions, including TPS, QPS, and XP-QPS. As these additional factors vary implicitly across runs, the surface captures the combined effect of XP-TPS and freshness under differing underlying system conditions, rather than an isolated causal relationship.

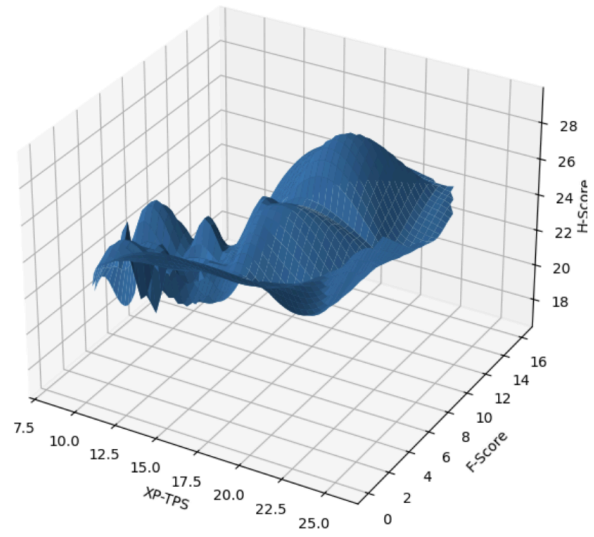


Figure 8. Impact of XP-TPS and Freshness on H-Score in 10x Runs

The highly irregular shape of the surface reflects this conditional dependency, which can also be seen in Figure 8 for the 10x scale runs. In several regions, higher XP-TPS does not translate into improved H-Score values, indicating that gains in transactional throughput can be offset by degradation in other contributing metrics, such as analytical throughput or baseline transactional performance. Similarly, improvements in freshness only yield higher H-Scores when accompanied by favourable behaviour in the remaining metrics.

At the 1x scale, where overheads and transient effects dominate, these interacting influences amplify variability, resulting in a fragmented performance landscape. Overall, the plot demonstrates that under low-scale conditions, the H-Score is governed by the interplay of multiple metrics, and improvements in XP-TPS or freshness alone are insufficient to guarantee holistic performance gains.

VI. Conclusion

Through a systematic benchmarking study with realistic banking workloads, our findings demonstrate that HTAP systems, when carefully configured and tuned, can consistently deliver sub-20ms data freshness while sustaining mixed OLTP and OLAP workloads at scale.

A key contribution of our work lies in the extension of the HyBench framework through the development of a Python-based benchmarking harness. Rather than treating benchmarking as a static, one-off exercise, the harness enables fully automated, repeatable, and parameterised experimentation across a wide configuration space. By orchestrating database restoration, housekeeping, cache prewarming, workload execution, and fine-grained metric collection, the harness provides a robust and extensible evaluation pipeline that goes beyond HyBench's native capabilities. This approach allows researchers and practitioners to explore performance trade-offs systematically, introduce custom metrics, and integrate domain-specific logic without intrusive changes to the DBMS or benchmark source code.

Equally important is the use of a banking-oriented schema modelled on real-world financial systems. HyBench's schema captures core entities, transactional patterns, and anomaly characteristics relevant to fraud detection, including high-frequency transfers, account state changes, and injected risk conditions such as blocked accounts and delinquent loans. This realism strengthens the external validity of the evaluation, ensuring that observed performance characteristics reflect practical deployment scenarios rather than synthetic or overly simplified workloads. The inclusion of freshness-aware timestamps and the F-Score metric further enables direct measurement of how quickly transactional updates become visible to analytical queries—an essential requirement for operational fraud detection.

Experimental results across HyBench's 1x and 10x scale factors confirm that HTAP systems can maintain consistently low freshness latency even under sustained mixed workloads. While unified execution incurs measurable throughput penalties compared to isolated OLTP and OLAP deployments, these trade-offs are offset by the elimination of ETL-induced data staleness, reduced architectural complexity, and the ability to perform analytics

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

directly on live transactional data. The results also highlight that default database configurations are insufficient for HTAP use cases; performance and stability depend heavily on informed tuning, workload-aware resource allocation, and proactive monitoring.

Nevertheless, this study also exposes practical limitations. At higher scale factors, hardware constraints, I/O saturation, and threading overheads can distort benchmark results, underscoring the importance of aligning benchmark scale with available infrastructure. Furthermore, contention between long-running analytical queries and latency-sensitive transactions remains a central challenge, reinforcing the need for improved workload isolation, adaptive scheduling, and HTAP-aware application design.

Looking ahead, the full adoption of HTAP architectures holds significant promise, but not without challenges. On the one hand, HTAP offers a path to simpler architectures, fresher data, and faster insights extraction, aligning closely with modern regulatory, operational, and customer expectations. On the other hand, widespread adoption will require rethinking application patterns, retraining engineering teams, and overcoming vendor lock-in as HTAP maturity varies across platforms. Moreover, while HTAP simplifies data pipelines, it shifts complexity into query optimisation, workload isolation, and configuration management; domains that still require advanced expertise and careful management. If these challenges can be addressed, HTAP could become a cornerstone for near-real-time, data-driven decision-making in finance and beyond.

References

[1] M.G. Giorgino and J.G. Vella, “Achieving Near Real-Time Data Freshness in Fraud Detection: An HTAP Approach,” DATA ANALYTICS 2025 : The Fourteenth International Conference on Data Analytics, 2025.

[2] O. O. Elumilade, I. A. Ogundeji, G. O. Achumie, and H. E. Omokhoa, “Enhancing Fraud Detection and forensic auditing through data‑driven techniques for financial integrity and security,” Journal of Advanced Education and Sciences, vol. 1, no. 2, pp. 55–63, 2021.

[3] D. Mangala and L. Soni, “A systematic literature review on frauds in banking sector,” Journal of Financial Crime, vol. 30, no. 1, pp. 285–301, 2023.

[4] N. Faisal, J. Nahar, N. Sultana, and A. A. Mintoo, “Fraud Detection in Banking Leveraging AI to Identify and Prevent Fraudulent Activities in Real‑Time,” Journal of Machine Learning, Data Engineering and Data Science, vol. 1, no. 01, pp. 181–197, 2024.

[5] E. A. Minastireanu and G. Mesnita, “An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection,” Informatica Economica, vol. 23, no. 1, 2019.

[6] A. Abdallah, M. A. Maarof, and A. Zainal, “Fraud detection system: A Survey,” Journal of Network and Computer Applications, vol. 68, pp. 90–113, 2016.

[7] S. A. Ionescu, V. Diaconita, and A. O. Radu, “Engineering Sustainable Data Architectures for Modern Financial Institutions,” Electronics, vol. 14, no. 8, p. 1650, 2025.

[8] Z. Zhang, A. Megargel, and L. Jiang, “Performance Evaluation of NewSQL Databases in a Distributed Architecture,” IEEE Access, vol. 13, pp. 11185-11194, 2025.

[9] C. Zhang, G. Li, J. Zhang, X. Zhang, and J. Feng, “HTAP Databases: A Survey,” IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 11, pp. 6410-6429, 2024.

[10] W. Lehner, “Merging OLTP and OLAP–Back to the Future: (Panel),” in International Workshop on Business Intelligence for the Real‑Time Enterprise, Springer, 2009, pp. 171–173.

[11] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, and B. Becker, The data warehouse lifecycle toolkit. J Wiley, 2008.

[12] D. Beyer, S. Löwe, and P. Wendler, “Reliable benchmarking: Requirements and solutions,” Int. Journal on Software Tools for Technology Transfer, vol. 21, no. 1, pp. 1–29, 2019.

[13] Gartner’s 2014 Hype Cycle for Emerging Technologies Maps the Journey to Digital Business – gartner.com [Accessed 01-01-2026]

[14] M. Alomari, M.J. Cahill, A. Fekete, and U. Röhm, “HTAP Databases: Bridging the Gap Between OLTP and OLAP,” in IEEE ICDE, pp. 1262-1267, 2014.

[15] R. Cole et al., “The mixed workload CH‑benCHmark,” in Proceedings of the Fourth International Workshop on Testing Database Systems, 2011, pp. 1–6.

[16] C. Zhang, G. Li, and T. Lv, “HyBench: A New Benchmark for HTAP Databases,” Proceedings of the VLDB Endowment, vol. 17, no. 5, pp. 939–951, 2024.

[17] C. Camilleri, C. Vella and V. Nezval, “HTAP with Reactive Streaming ETL,” JCIT, vol. 23, no. 4, pp. 1-9, 2021.

[18] J. Yang, et al., "Twisted Twin: A Collaborative and Competitive Memory Management Approach in HTAP Systems," in Proceedings of the VLDB Endowment vol. 18, no. 10, pp. 3312-3325, 2025.

[19] A. Kemper, and T. Neumann, "Hyper: Hybrid oltp&olap high performance database system," in IEEE 27th International Conference on Data Engineering, pp. 195-206, 2010.

[20] F. Coelho, J. Paulo, R. Vilaça, J. Pereira, and R. Oliveira. “Htapbench: Hybrid transactional and analytical processing benchmark,” in Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering, pp. 293-304, 2017.

[21] E. Milkai, Y. Chronis, K.P. Gaffney, Z. Guo, J.M. Patel and X. Yu, “How Good is my HTAP System?” in SIGMOND, ACM, pp. 1810–1824, 2022.

[22] D. Bitton, D.J. DeWitt, and C. Turbyfill, “Benchmarking database systems: A systematic approach,” Computer Sciences Department, University of Wisconsin, 1983.

[23] M. Poess, “New initiatives in the TPC,” in Technology Conference on Performance Evaluation and Benchmarking pp. 127-148, 2022.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Confirming the Trustworthiness of a Credential Issuer Using Remote Attestation

Rainer Falk and Steffen Fries
Siemens AG
Foundational Technologies
Munich, Germany
e-mail: {rainer.falk | steffen.fries}@siemens.com

***Abstract*—Authentication credentials, e.g., device authentication certificates, may be issued locally in industrial environments. However, a locally issued credential may not be as trustworthy as credentials issued by a highly protected centralized security infrastructure. An attestation can confirm evidence of its trustworthiness. Including such an attestation of the credential issuer within an issued authentication credential allows a relying party to check this information as part of credential validation. This paper proposes to embed such a cryptographically verifiable integrity attestation of a certificate issuer into issued authentication certificates. It allows verifying whether the credential issuer was in a trustworthy state when it issued that authentication credential.**

Keywords–cybersecurity; attestation; credential; digital certificate; device authentication, industrial security.

I. INTRODUCTION

Authentication credentials, e.g., digital certificates or authentication tokens, allow a user to authenticate, i.e., to prove a claimed identity in a cryptographically protected way. Credentials can be issued by a highly protected issuer like a Certification Authority (CA) of a Public Key Infrastructure (PKI), following well-defined operational security processes, or by an Identity and Access Management (IAM) service like for instance an Open Authorization (OAuth) [2] authorization server. However, also other deployment options providing local independence and increased flexibility are used in Operation Technology (OT). Digital certificates or authentication tokens may, e.g., be generated by engineering tools, by a device management system, locally on industrial devices implementing an embedded CA (also called alias CA), or by a service of an industrial edge system. This supports a self-contained operation, independent of backend connectivity. The execution environments that create such credentials may therefore have different technical protections, leading to different levels of trustworthiness compared to a centralized security infrastructure that may be deployed in a physically protected security data center, which is operated and maintained following relevant security standards. Some of the used execution environments might be manipulated, e.g., if a vulnerability in the implementation is exploited. Furthermore, the program code creating authentication credentials may be updated with less strict controls compared to centralized security infrastructures.

This paper, as an extended version of [1], describes how to include a cryptographically protected attestation of the credential issuer within an issued digital end entity certificate. The attestation confirms the integrity of the issuer's execution environment at the point in time when the digital certificate was issued. The cryptographically protected attestation confirms the actual integrity evidence of the used execution environment, confirming that the issuer was in fact in a trustworthy, uncompromised state when it issued this credential, running valid software code. Including the attestation within an issued authentication credential allows verifying the integrity and thereby the trustworthiness of the execution environment and the executed program code in which a credential has been created. The issuer attestation corresponds to the point in time when the authentication credential has been created. It can be verified and validated at a later point in time when the authentication credential is used, i.e., when validating the authentication certificate. The trustworthiness of an authentication credential is therefore determined depending on the included issuer's integrity attestation (platform attestation).

This approach is specifically promising if a centralized, implicitly trusted PKI is not, or at least not permanently, available in an operational environment, e.g., an industrial automation and control system. In such cases, a local CA may be deployed and operated on a device or within a local part of an automation system. The integrity attestation of the issuing device can provide an increased level of trustworthiness for generated credentials, as the attestation functionality that creates the attestation can be protected at a higher level than the functionality to which the attestation relates, in particular if a hardware-based attestation implementation is used.

The remainder of the paper is structured as follows: Section II provides an overview on boundary conditions given by industrial security requirements and on technical considerations when issuing credentials. Section III introduces the concept of providing a statement of the security status of the credential issuer execution environment, allowing a relying party to determine trustworthiness in the issued certificates. Section IV outlines different approaches to verify the issuer's attestation of a credential. Section V outlines an industrial usage example, a substation of an energy automation system including a local CA. Section VI contains a preliminary evaluation, and Section VII concludes the paper and gives an outlook towards future work.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

II. Related Work

This section provides an overview of relevant related work.

A. Industrial Security

Protecting Industrial Automation and Control Systems (IACS) against intentional attacks is demanded by operators to ensure a reliable operation, by industrial security standards as IEC 62443 [3], and increasingly also by regulation [4][5]. Security requirements defined by the industrial security standard IEC 62443 range from security processes during development and operation of devices and systems, personal and physical security, device security, network security, and application security, addressing the device manufacturer, the integrator, as well as the operator of the IACS. IEC 62443 specifically describes technical requirements on system and component level, targeting four different security levels, which relate to the strength of a considered attacker. Moreover, this framework also contains specific requirements regarding authentication methods and credentials, as well as the use of cryptographic algorithms including their strength.

Industrial security is also called OT security, to distinguish it from general Information Technology (IT) security. In OT systems, actions in the digital world typically have a direct impact on the physical world. Therefore, industrial systems have different security priorities and requirements compared to common IT systems. Typically, availability and integrity of an automation system have higher priority than confidentiality. Specific requirements and side conditions of industrial automation systems like high availability, planned configuration (engineering info), scheduled maintenance windows, long life cycles, unattended operation, real-time operation, and communication, as well as safety requirements have to be considered when designing an OT security solution. Such side conditions, coming from the operational environments and use of OT systems, remain valid also with increasing IT/OT convergence, the increasing opening towards external systems as business systems, cloud services, and data exchange between business partners in ecosystems.

B. Cyber Resilience

Risk management, the established approach to cyber security, identifies threats and determines the risk depending on probability and impact of a potential attack in a dedicated operational environment. The objective is to put the focus of defined security measures on the most relevant risks, reducing the probability that a successful attack takes place, and reducing the impact of successful attacks, e.g., by detecting successful attacks by security monitoring allowing to react, e.g., by shutting down a Cyber Physical System (CPS).

Cyber resilience puts the focus on a reduction of the impact of successful attacks, where the system can stay operational with a degraded performance or functionality, and to recover quickly and reliably from a successful attack. Under the assumption that successful attacks can and will happen, the system is designed in a way that attacks can be handled in a reasonable way, limiting the potential impact on operations. This may lead to shutting down only parts of a CPS that are directly affected by an attack, but keeping other functionality running. An example may be a protection device in a substation of a power distribution network. In case of a denial-of-service (DoS) attack on the communication interface, the protection device may switch off the communication interface, but still retain the functionality to trip power in case of a detected electrical anomaly. The main functionality in this case is the protection to ensure availability and safety. Due to the DoS attack on the communication interface, the reporting communication of measurement values to a substation controller or control center would not be possible during the attack, resulting in a reduced functionality.

Cyber-resilience gets increasing attention, as can be seen by recent security standards and the regulation of the European Cyber Resilience Act (CRA) [4]. The standard NIST SP800-193 [6] describes technology-independent guidelines for resilience of platform firmware.

C. Authentication Credentials

Authentication credentials are used to confirm the identity of a user (human user, software process, or device) towards a relying party. Examples are, besides passwords, authentication tokens, also digital authentication certificates. A digital certificate is an authentication credential that binds the public key of a user to the user's identity. A digital certificate can include also a certificate practice statement that provides information on the trustworthiness of the issuing process as specified in [6]. A widely used certificate format in IT and also OT domains is X.509, specified by the International Telecommunication Union (ITU-T) [8].

A digital certificate is typically issued by a CA of a centralized security infrastructure, i.e., a PKI infrastructure. A digital certificate may be issued also by a CA that is realized as part of an engineering tool, a device management tool, a local security server, be integrated on a device (embedded CA). Alternatively, self-signed certificates directly generated on the device or externally by a tool may be used. Standards like Trusted Computing Group's TCG specification "Device Identifier Composition Engine" (DICE) [9] and Desktop Management Task Force (DMTF) specification "Security Protocols and Data Models" (SPDM) [9] define that a device can include an internal CA for issuing device certificates, called "embedded CA" or "alias CA". A device-local CA allows a device to issue a device certificate for itself that includes information on changeable device information, in particular on its current firmware version that changes when installing a firmware update. Such a device certificate does identify not only a static device identifier (e.g., manufacturer, model, serial number), but also information on the device that changes during the lifecycle of the device. Besides the current firmware version, also a device lifecycle state (e.g., factory default, provisioned), or an information on the current device configuration, e.g., an identifier of the device configuration settings or a version information, may be included.

Certificate transparency, specified by IETF RFC 9162 [11], allows including Transport Layer Security (TLS) server certificates issued by a PKI in a public certificate log, called certificate transparency log. It allows, e.g., the owner of an Internet domain to monitor all certificates that are issued by a CA for the owner's domain. Thereby, the owner can detect if

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

a digital certificate is illegitimately issued for the owner's domain, i.e., indicating as certificate identity (subject) the owner domain, while the certificate has not been requested by the domain owner, but rather by someone else. In certificate transparency, a digital certificate can comprise an inclusion proof as certificate extension. It confirms reliably that the issued certificate has in fact been included in a certificate transparency log. This supports auditing of issuing CAs to detect if a CA issued certificates that were not intended by the operating organization, and to audit that issued certificates are in fact put as expected into a certificate transparency log by the issuing CA.

D. Remote Attestation

A remote attestation is a cryptographically protected data structure that can confirm security-relevant information called evidence about a device (platform attestation) or of a cryptographic key (key attestation). The Remote ATtestation procedureS (RATS) architecture specification IETF RFC 9334 [12] gives an overview on remote attestation use cases. It describes also different approaches to ensure freshness of an attestation, allowing to verify when a provided attestation is in fact up-to-date representing a current device information, i.e., it has been created recently. If trusted, synchronized clocks are available, a timestamp can be included in the attestation, allowing to determine the point in time when the attestation was created. Alternatively, a nonce, an unpredictable (random) value is provided by a verifier in a request, that is to be included in the signed attestation. It ensures that the signed attestation received as response, including the nonce, has been created after the nonce has been received, binding it to the time when the request was sent. However, the nonce allows only the verifier who determined the nonce to verify the freshness. A third approach is to include an (unpredictable) epoch IDentifier (epoch ID) provided by an epoch ID distributor in the attestation. An epoch ID may be understood to be similar to a "global nonce" that changes unpredictably over time, independently of a specific request.

Meanwhile, standardization has started to include remote attestation also in the process of requesting certificates using different formats [13], which can be directly used in typical certificate enrollment protocols. The defined extension allows conveying evidence and attestation results within a certification request. This in turn enhances the verification options of the issuing CA (or rather its RA validating a received certificate signing request) beyond the common verification of proof-of-possession of the private key corresponding to the public key in the certificate request, and the proof-of-identity of the requestor, towards a protected confirmation about the platform properties of the platform that generated the certification request, i.e., confirming the integrity and trustworthiness of the requestor. Once provided as part of a certification request, the attestation statement for the requestor may also be included in the issued certificate for later verification by the relying party.

Note that the focus of this paper is not the integrity attestation of the requestor of a certificate, but the integrity attestation of the certificate issuer. Both attestations can be included, allowing to attest properties of both the requester (e.g., a key attestation as statement how a keypair was generated and its platform integrity), as well as security-relevant properties of the issuer's execution environment, confirming the integrity and trustworthiness of the certificate issuer when it issued the certificate.

III. Issuing Credentials Including an Issuer Attestation

A digital certificate, as proposed in this paper, can include a cryptographically protected attestation that confirms the issuer's integrity at the point in time when the digital certificate was issued. The cryptographically protected attestation confirms the actual integrity evidence of the execution environment of the issuer. Including the attestation within issued digital certificate allows verifying the integrity of the execution environment in which the certificate has been created. The trustworthiness of a digital certificate can therefore be determined depending on the included issuer's integrity attestation. This can be beneficial specifically if the certificate issuer is not in a centralized protected environment, but on an industrial site, presumably without permanent connection to a central infrastructure. In this setup, it has to be assumed that the certificate issuer itself could be manipulated. The attestation allows a peer validating the certificate to check the issuer's integrity status at the point in time when the digital certificate was issued. If the issuer's execution environment (platform and running software) would have been manipulated when issuing a certificate, this could be detected later, allowing to reject the issued certificate as invalid and also to identify to component as potential root cause for a malfunction. This approach enhances cyber-resilience as the impact of issuing certificates by a manipulated local CA can be limited.

A. Digital Certificate Including Issuer Attestation

A digital certificate binds a public key to the identifier of a subject (e.g., human user, device, process). It is signed by the issuer, e.g., a CA. It is proposed to include in addition an attestation to the issued digital certificate that confirms the integrity of the certificate issuer. For X.509 certificates, this can be realized by using the extension capability of certificates, allowing to include additional information by the issuer within an additional certificate extension field.

Figure 1. Digital certificate including an integrity attestation of the issuer.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Figure 1 shows the main conceptual elements of a digital certificate that includes the issuer's attestation in addition to the subject's identity and public key. Note that further certificate components that may be included, like the validity period or the key usage, are not shown. The issuer attestation would be signed by an attestation key of the issuer's attestation component, while the digital signature of the issued certificate would be signed by the private signing key of the issuing CA. This is seen as specifically useful if the issuer is a so-called embedded CA included on a device (embedded CA). The attestation included can be a platform attestation that allows verifying the integrity of the embedded CA issuing the digital certificate, as well as a key attestation confirming the key store type, e.g., a secure-element-based key store, of the issuer's key store, i.e., of the private key used to sign the issued digital certificate.

The attestation included in the device certificate allows verifying the integrity status of the device, in particular of its embedded CA component, that was active at the point in time when the device certificate was issued. The attestation includes evidence information on the device platform. Commonly included evidence may confirm that secure boot is in place, the loaded boot loader and the applied boot configuration, the loaded operating system, and also the started programs. Evidence may confirm as well whether a software component is in fact executed within a specific, isolated Trusted Execution Environment (TEE). The evidence is determined by a trusted measurement component, which may be called root of trust for measurement. The attestation included in the device certificate is signed by an attestation key of the issuer. A main idea of attestation is to realize a highly protected attestation functionality, e.g., protected by hardware mechanisms, that can provide a reliable information on software that is executed in a general execution environment.

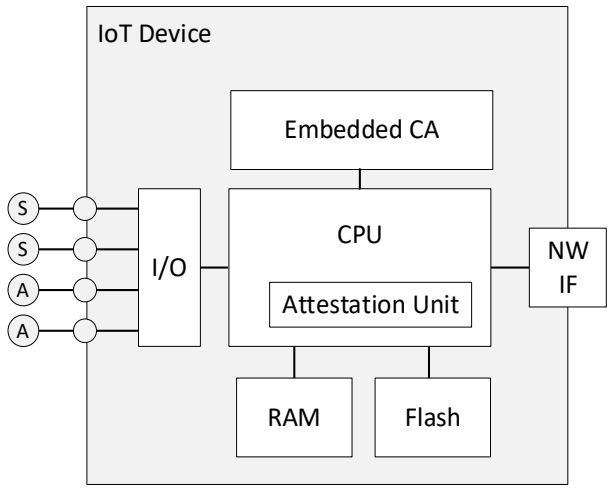


Figure 2. IoT device including an embedded CA.

Figure 2 shows an Internet of Things (IoT) device, e.g., an industrial control device, that includes an embedded CA for issuing digital device certificates. A device certificate issued locally by the embedded CA may include identifying information, such as the device type and serial number, as well as the currently installed firmware version or the loaded device configuration.

The embedded CA on a device can be realized, e.g., by a dedicated secure element, a logically isolated enclave, or just an (isolated) software component / app running on the device. This realization information may also be part of the attestation statement, which allows a relying party to make a more fine-grained decision about the issuer's trustworthiness. The device would include furthermore an Attestation Unit (AU), involving a measurement component to determine evidence in a trustworthy way (root of trust for measurements), and a component for issuing the attestation based on the determined evidence. Such attestation functionality is supported on common compute platforms, e.g., using a secure element that can be integrated within the CPU, as shown in Figure 2, or by a dedicated hardware component.

B. Issuing Process Adding Attestation Information

When a digital certificate including an issuer's attestation is to be issued, an attestation concerning the certificate issuer has to be determined and included on the digital certificate. So, the determined attestation can be added to the issued certificate as part of the certificate issuing process.

The Certification Unit (CU) comprises the Registration Authority (RA) and Certification Authority (CA). It includes also the Attestation Unit (AU) with its Attestation Measurement Unit (AMU) and the Attestation Signing Unit (ASU). The CU may be an internal component of a device featuring an embedded CA, or may be part of an external CA (e.g., in an engineering tool or standalone). The CA uses a Hardware Security Module (HSM), e.g., a crypto controller, to create the digital signature of the certificate (Cert) that is then provided to the requesting device.

In the example message sequence shown in Figure 3, the RA extends the Certificate Signing Request (CSR) received form the device with the attestation determined by AU to create the "to-be-signed certificate" ASN.1 data structure (`tbsCert`). The RA sends the extended CSR to the CA for signing, resulting in the signed certificate including the CU's attestation.

First, the device generates its key pair and the corresponding certificate signing request (CSR) and sends it to the CU's RA. The RA obtains the CU's attestation (AttCu), from the AU, and extends the received CSR accordingly by adding the CU's attestation (AttCu) as extension to the "to be signed certificate" (`tbsCert`). The attestation includes evidence depending on the measurements that have been obtained by the AMU. The measurements are usually collected before the attestation is built (as shown in Figure 3). However, it is also possible to determine some measurements on demand, i.e., after the attestation has been requested.

The measurements may cover information on the CU's components (RA, CA), e.g., the software version and integrity information of the compute platform on which they are executed. Examples are information on whether secure boot has been active during start-up, and integrity information of the loaded and executed operating system and its components. Measurements are included as evidence within an attestation.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

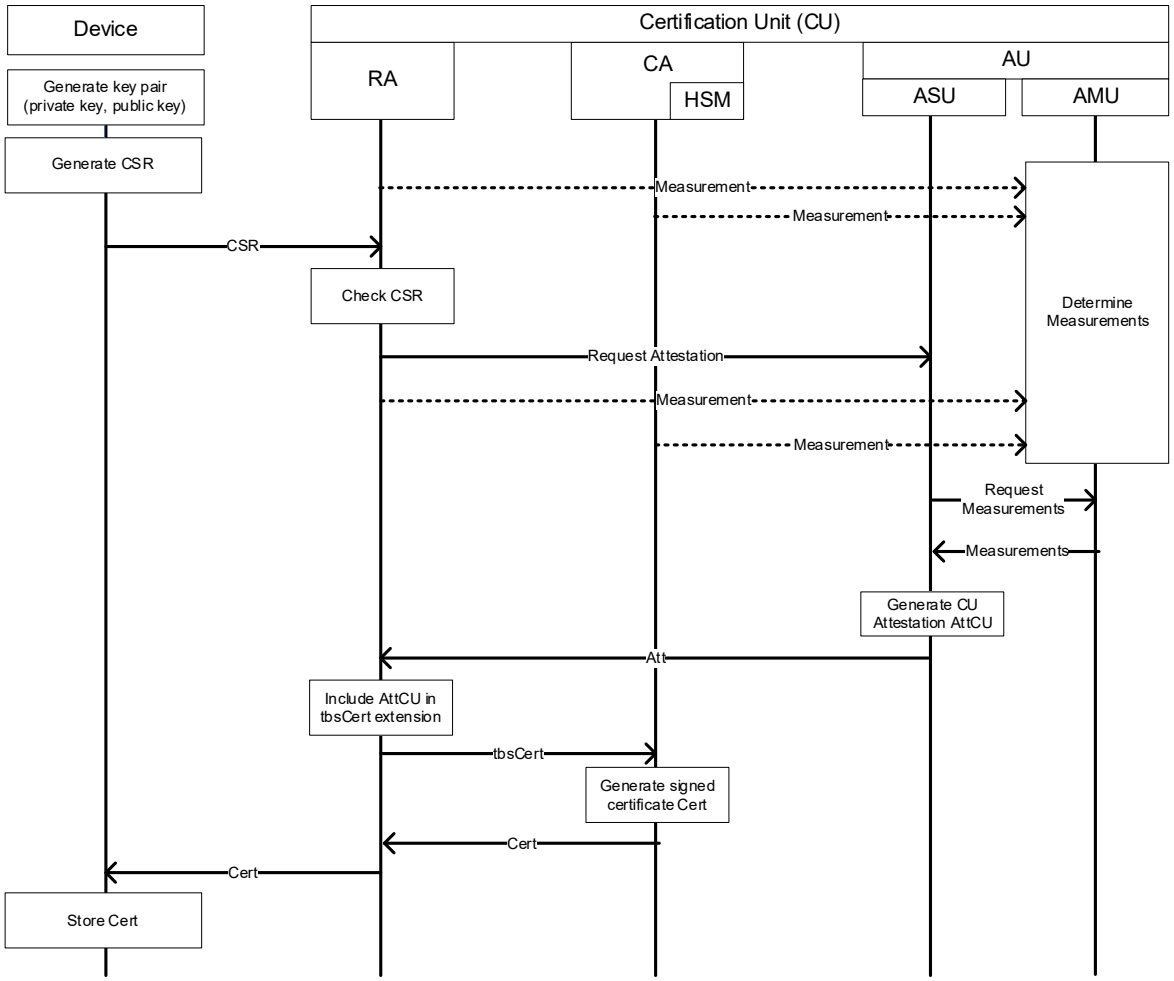


Figure 3. Including an attestation during credential issuing.

It is also possible to attest that certain software components, as here RA and CA, are in fact executed in a specific isolated, highly protected execution environment, in particular in a specific confidential computing environment. Confidential computing refers to technologies where memory content is encrypted during processing (data in use protection), the evidence confirming that a software component is in fact executed within such a protected execution environment.

Validating such evidence information allows to determine whether the CU can in fact by trusted by an *external* party. Such information is complementary to manual audits that are performed for centralized PKIs, e.g., on a yearly basis. Including such information in issued certificates allows even the parties validating the certificate to check whether the CA that issued this certificate was in fact in a trustworthy state at the time when this specific certificate was issued.

Besides the included information about the platform itself, the attestation may also contain freshness information. This allows the relying party to verify that the attestation was in fact provided as part of the issuing process, i.e., that it is not stale information that has been residing on the CA for a longer time period.

Freshness information of an issuer attestation may be provided in different ways:

- Usage of timestamps (real-time clock, or a synchronized local time information). This requires that a trustworthy, synchronized time source is available within the target operational environment.
- Application of a nonce, an unpredictable one-time value, provided by the certificate requestor. This nonce may be provided as part of the certification request as outlined in [12] and [13]. Unfortunately, only the requester who determined a nonce value can verify whether a certain nonce value is in fact recent. Therefore, rather an epoch ID provided by a trusted epoch ID distributor may be used. In this case, it would be additionally required to record during which time period a certain epoch ID was valid, to enable checks at later point in time (and requiring the implementation to ensure that only the current epoch ID is applied).

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

- Furthermore, a hash value created deterministically from certificate content may be used, binding the attestation to the issued certificate. It may be applicable if the certificate includes an unpredictable information, in particular a public key known to be freshly generated, or a time stamp information as the beginning of the certificate's validity period that is known to be the time of certificate issuing. As outlined in IETF RFC 3161 and IETF RFC 5280, a trusted time can be included in certificates, which can be obtained by a trusted time service.

Which freshness approach is best suited depends on the specific deployment scenario and the available infrastructure and may be determined by an operator's security policy.

It may be advantageous to use information of the certificate request to bind the attestation to the specific certificate, e.g., serial number and start of the certificate's validity period (certificate field "`notBefore`"), or the included public key and public key algorithm. This approach to determine a freshness value for the attestation based on information included in the certificate/credential request may be called "freshness steeling", i.e., to deterministically determine an attestation freshness value depending on the information included in a certificate/credential request. If certificate transparency is in use, the freshness of the attestation and the binding to a certificate may be supported by including an issued certificate including the issuer attestation as extension immediately in the certificate transparency log.

C. Validation a Certificate Including an Issuer Integrity Attestation

A device can provide its digital certificate including the additional attribute (extension) including the issuer's attestation when authenticating towards a communication peer, e.g., as part of

- any security protocol handshake utilized by the device that employes certificates. A prominent example is the authentication and key agreement in Transport Layer Security (TLS, IETF RFC 8446 [16]),
- network attachment to provide the device certificate as data element, e.g., towards a device/network management system, which may be done in the context of an IEEE 802.1x [17] based network access authentication,
- application-level protocols or data exchanges utilizing digital certificates.

The relying party validates the received digital certificate following the validation rules specified in ITU-T X.509 [8], which include, e.g., the verification of the subject name, validity period, certificate revocation status, but also the integrity of the digital certificate itself. Besides the digital certificate itself, also the certification path from the digital certificate to a trusted root certificate is validated (certificate path validation). This certificate path validation involves validating the certificate of the issuing CA. For acceptance, the results have to comply with a defined security policy.

The inclusion of an attestation statement of the issuing CA in the device certificate allows additionally to match the trustworthiness of the issuing CA to an expected state necessary for processing certain data. As indicated before, this is specifically interesting for embedded CAs, i.e., a device-local CA issuing a device certificate including changeable device information as the device's current firmware version. It may be useful also for a local CA that issues certificates for devices within a local deployment, e.g., for the devices within a substation of an energy automation system. Such a local CA may be useful to ensure continued operation within a local deployment environment independently of whether access to backend systems, typically residing in a control center of the power system operator, is currently available or interrupted. The approach has been described for a CA issuing end entity certificates, but could also be applied for intermediate CAs issuing certificates for a subordinate CA.

Depending on the trust evaluation based on the attestation included in a received authentication certificate, a relying party may decide to, e.g.,

- limit the authoritative actions the certificate holder may perform (i.e., limit the granted permissions),
- perform additional plausibility checks on data received from the device,
- provide only uncritical or non-sensitive information to the device, or
- reject interaction completely if the certificate based on the issuer information does not match the expected trustworthiness expectation.

The attestation included in a digital certificate enables therefore a more specific interaction with a device depending on its own state but also depending on the state of the issuing CA at the time of issuing the certificate. Even in the case that an issuing CA should be compromised, it can be detected reliably and reacted on accordingly.

IV. Verifying Authentication Credentials Including an Issuer Attestation

Different approaches can be taken to verify the issuer's attestation included in an authentication credential:

- The authentication credential can be accepted as valid only if the included issuer's attestation can be validated successfully. The credential is thereby only valid if it can be verified that the credential issuer was in a trustworthy state when it issued this credential. In addition to verifying the credential's signature, the digital signature (or cryptographic checksum, in general) of the included issuer attestation is valid, and whether the evidence included in the issuer attestation complies with a given attestation validation policy.
- The authentication credential may be accepted independently of the included attestation (i.e., as if no attestation would be included). But the attestation may be logged, allowing for monitoring in backend systems or during audits whether the issuer of peer authentication credentials was compromised when issuing certificates. This would have an impact on the credential validation time, as the issuer attestation would only be stored and not immediately verified. This may be beneficial if a credential is used in a key agreement handshake in a security protocol like TLS [16]. The authentication

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

credentials used may be provided by communication peers on their own, or using network monitoring using a network monitoring system that intercepts authentication and key agreements (as far as authentication and key agreement exchanges are transmitted without being encrypted).

- It may be possible as well that permissions associated with the authentication credential are granted depending on the attestation included in the authentication credential. This means that depending on whether the issuer attestation is valid, a node may receive limited access permissions.
- A device may validate the issuer attestation of its own authentication certificate when it receives it from the issuing CA. Depending on the validation of the included issuer attestation, the device may accept and store the received certificate.

Therefore, the attestation may be, depending on the usage environment, used in different ways, allowing also for a smooth introduction in existing systems.

V. USAGE EXAMPLE

As an example from an industrial application use case, a protection device of a substation of an energy automation system may be considered as control device. Protection devices are applied within electric power systems to detect abnormal and intolerable electrical conditions and to initiate appropriate corrective actions, e.g., to interrupt a power line. Protection devices of a substation can communicate their measurements to other substation devices in the substation over a cryptographically protected communication link. The digital certificates used for authentication by the protection devices of a substation can be issued by a CA being part of the substation. This allows the substation to operate autonomously even if no backend connectivity is available, or when backend systems should not be operational.

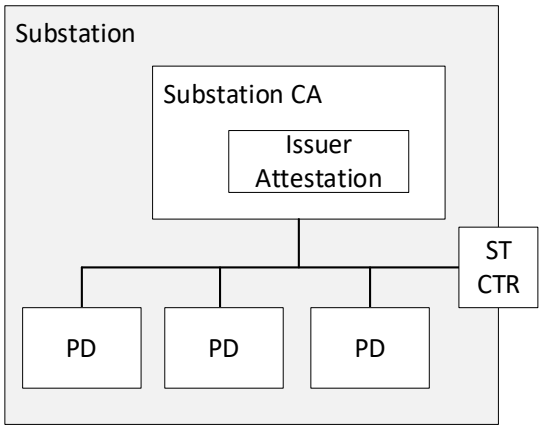


Figure 4. Substation with local CA.

Figure 4 shows, as application example, a substation with three Protection Devices (PD) that communicate via a substation controller (ST CTR) with other components of an energy automation system. The substation includes a substation CA that issues authentication certificates for the protection devices and also the substation controller, enabling authenticated and cryptographically protected network communication within the substation. The substation-local CA enables an autonomous, self-contained operation of the substation, i.e., independently of a centralized security infrastructure. Even if network communication with the energy automation system's control center should be temporarily interrupted, the substation can continue its operation autonomously.

The substation's CA may include its attestation in digital authentication certificates issued to the protection devices of the substation. Protection devices can validate the attestation included in authentication certificates provided by other protection devices or the substation controller of the substation. Depending on the result, communication with a protection device providing an authentication certificate including an invalid attestation may be rejected, or the measurement values and commands of the peer protection device may be restricted, limiting the possible impact that can be caused by the peer protection device.

VI. EVALUATION

This section gives a preliminary evaluation of the presented concept from different perspectives.

CPS availability perspective: It is possible that an authentication credential is accepted for setting up an authenticated communication session independently of the included issuer attestation. This ensures that CPS availability is not reduced by the proposed extension. But still, it allows for recording and logging the information enabling later analysis. If, however, the attestation is validated when setting up an authenticated communication session, the proposed approach allows maintaining CPS operation in a limited way even when the issuing CA should have been manipulated. A reliable CPS operation can be maintained, avoiding the need to shut down the CPS operation completely. A device not having an authentication certificate including a valid issuer attestation may activate a limited function mode, providing continued operation of the CPS if no trustworthy CA should be available. Granted access permissions or the allowed value range of control parameters can be reduced to limit the possible impact of a successful attack affecting the CA.

Performance perspective: Creating an attestation causes some computational overhead. Also, the issuer attestation enlarges the size of authentication certificates. So, the approach leads to a certain overhead. However, in a CPS, the overhead affects only the issuing and validation of authentication credentials, not the real-time communication taking place after a secure, authenticated connection has been established.

CPS management perspective: The operator of the CPS has to be aware about the issuer attestation functionality. Also, the operating personnel has to be trained for the functionality (e.g., failure diagnostics, maintenance). To simplify usage, the details of the underlying attestation technology can be shown to operators and service technicians in an abstracted way, e.g., indicating a limited trust category, and proposing suitable reactions, e.g., to install a recent firmware update, to trigger a

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

re-issuing of locally used certificates, and to perform a test of the correct functionality.

Implementation perspective: A CA as well as control devices have to implement the proposed functionality. The implementation overhead depends on the specific technical implementation approach. In particular, the CA has to be extended to support attestation, i.e., to perform trustworthy measurements and to create a corresponding attestation artefact. Control devices have to be extended to record/log or to validate the issuer attestation. The complexity depends in particular on the granularity of evidence included in the issuer attestation, as fine granular evidence may require complex validation.

Overall, implementing, engineering, and testing the proposed extension implies additional effort that has to be justified, in particular by the reduced impact that can be caused by a manipulated (local) CA. This benefit depends on the risk exposure of the overall system in which the extension is used: It may become more and more relevant when systems with local CA functionality that have in the past been deployed and operated in isolated, protected environments, are opened up to communicate with external systems in which they are exposed to an increasing threat landscape.

This risk of a manipulated (local) CA can be reduced by limiting the potential impact, as the issuer attestation allows detecting such a case reliably, and to react accordingly. The risk reduction that can be achieved by the proposed issuer attestation can be evaluated using a Threat and Risk Analysis (TRA). A TRA is typically conducted at the beginning of the product or system design or system development, and updated after major design changes, or to address a changed threat landscape. In a TRA, possible attacks (threats) on the system are identified. The impact that would be caused by a successful attack and the probability that the attack happens are evaluated to determine the risk of the identified threats. The risk evaluation allows prioritizing the threats, focusing on the most relevant risks and defining corresponding security measures. Appropriate security measures can target reducing the probability of an attack by preventing it, or by reducing the impact of a successful attack.

As long as the technology proposed in the paper has not been proven in a real-world operational setting, it can be evaluated conceptually by analyzing the impact that the additional security measure would have on the identified residual risks as determined by a TRA. However, TRAs for real-world CPS are typically not available publicly.

To illustrate the risk reduction, the threat of a manipulated local CA is considered. A realistic TRA would usually include a rather long list of threats. The likelihood and the impact of the threat is determined by judgement of competent personal, usually in a team including technical experts and people responsible for the product or system, and preferably also people involved in operation. To ensure consistency, criteria are defined that specify the conditions to assign a certain category. It has shown to be useful to define and document explicitly the criteria leading to the categorization of likelihood and impact, including also the made assumptions on the operational environment to ensure consistency and to allow for review. The TRA with prioritized risks is the basis for security design decisions, focusing on the most critical risks. It is the basis to define a security concept that includes suitable protection measures. The residual risk has to be accepted by management decision.

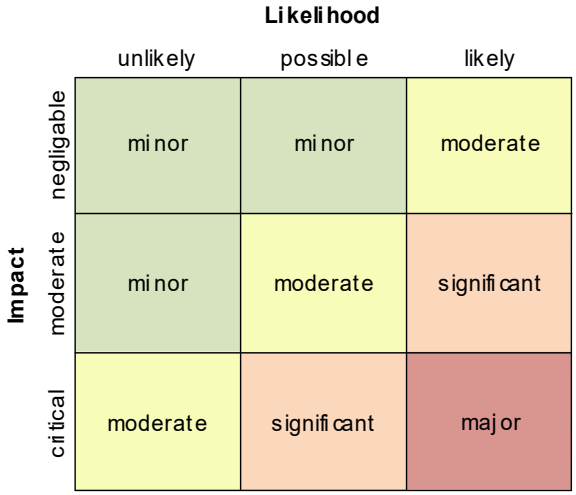


Figure 5. Risk mapping based on likelihood and impact.

The corresponding risk is determined based on the determined likelihood and impact, see Figure 5. It shows how the determined likelihood and the impact categories can be mapped to the corresponding risk category. In the example, the three categories unlikely, possible, and likely are used to describe the likelihood. For the impact, the three categories negligible, moderate, and critical are used. They are mapped to the corresponding risk rating (minor, moderate, significant, major). Minor and moderate risks may be acceptable, while risks rated as significant or major have to be addressed by suitable countermeasures, reducing the resulting risk to an acceptable level.

In industrial practice, also more fine-granular rankings can be used, distinguishing, e.g., four or five different categories for likelihood and impact. Also, the risk evaluation can in general include further categories, e.g., disastrous or negligeable.

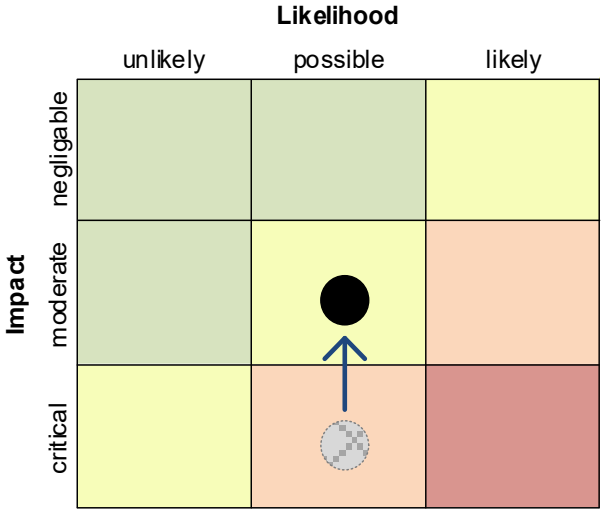


Figure 6. Risk reduction for the example threat.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

As example, the likelihood of a corruption of a local CA implementation is assumed to be rated as "possible", the impact being categorized as "critical". This leads to a "significant" risk, see Figure 6 (grey bubble).

If the likelihood of the threat cannot easily or practically be reduced in the application scenario, considering the relevant side conditions such as, e.g., brownfield components or usability by OT personnel, the alternative direction is to reduce reliably the possible impact that can happen if the attack should take place. Technical security controls may reduce the need for organizational or personnel security measures that come with operational overhead. In the example shown in Figure 6, using an issuer attestation reduces the impact from critical to moderate, leading to a "moderate" risk rating (black bubble), moving it to an acceptable risk category.

VII. Conclusion and Future Work

The concept described in this paper enhances authentication credentials with a reliable statement confirming the credential issuer's platform security state. It confirms whether the credential issuer was in fact in a trustworthy state when issuing an authentication credential. During validation, it can be matched with an operator's expectations regarding the trustworthiness of the certificate issuer. The approach allows a more fine-grained reaction based upon the attestation statement, e.g., limiting permissions that are effectively granted towards o the entity authenticated by the authentication credential.

It is planned to further evaluate the approach from a theoretical and practical perspective, including how it contributes to enhanced cyber-resilience in cyber-physical systems [18]. As part of the conceptual analysis, the expected overhead, relevant attack scenarios, and the limitations can be evaluated. Further work is needed to determine how to deal with different attestation validation results. Besides rejecting a certificate, more specific reactions could be triggered, e.g., limiting associated access permissions, or planning a maintenance action as, e.g., replacement of the affected device. A prototypical implementation, yet to be implemented in future work, can support the evaluation of the performance overhead (e.g., increased size of certificates including attestation information, added latency both for issuing and for validating such a certificate including an attestation that has to be validated as well). Furthermore, specific scenarios of a compromised issuer can be evaluated, in particular for an issuer which security configuration is not compliant with an expected policy, and for a compromised issuer. It allows evaluating practically which scenarios can be detected based on the certificate issuer's attestation included in the certificate. The proposed approach relies on the property that the attestation functionality that creates the attestation is protected at a higher level than the credential issuing functionality to which the attestation relates. It can be evaluated in future work which attack scenarios would lead to a compromised attestation functionality, making the limitations of a particular attestation technology transparent. Such evaluations are the basis for deciding how well it fits a specific target environment. Which approach fits for protecting freshness of the attestation depends on the specific deployment scenario and the available infrastructure. As in industrial automation systems, often, no, or at least no reliable, real-time clock is available, other options, as outlined in Section III are alternative candidates.

Adding an issuer attestation directly to issued credentials has the advantage that a communication peer validating a credential can directly validate also the credential issuer's trustworthiness. An alternative could be to use a dedicated integrity monitoring system for the CA, and putting the issued certificates on a negative list if the CA should be detected as compromised. Note that a regular certificate revocation may not always be possible in this case, as specifically in the device local CA approach, the issuing CA itself, which is assumed to be manipulated, would have to do the revocation after the manipulation has been detected. If a centralized (hierarchical) PKI is used, the superordinated CA would revoke the compromised CA's certificate. Furthermore, the CA could have issued certificates for an unknown time period until its manipulation is detected, leading to an uncertainty which certificates need to be revoked.

Adding an issuer attestation to a certificate may be done in addition to certificate transparency [11], i.e., both approaches can be combined in a single solution. Monitoring issued certificates and the included issuer's attestation would allow third parties furthermore to detect, independently of the actual usage of an issued certificate, if a credential issuer is not compliant anymore or if it becomes compromised. The transparency covering additionally the CA's integrity would allow third parties to independently monitor their trust state. The comparison of such approaches and also of their combined usage, and the achievable benefit, is a further area for deeper investigation.

Evaluating attack scenarios practically in a real-world CPS may require significant effort, and critical attack scenarios can usually not be tested directly in operational systems. A CPS simulation environment or specific testbeds can be used instead to determine the impact of threats even in a complex CPS setting. In a simulation environment, the actual impact of successful attacks can be evaluated by simulating relevant attack scenarios, determining the impact on the control system and on the physical domain.

An important objective of future work is to improve CPS cyber-resilience, allowing the CPS to stay operational under certain attack scenarios. A responsible, trustworthy CPS operation can be maintained even under certain attack scenarios if the impact of a successful attack can be reliably ensured to stay limited. Having reliable information on the actual integrity status of CPS components even under ongoing attacks allows keeping an attacked CPS operational in a trustworthy way, at least as long as it can be verified that the observed attack scenario is covered by the implemented cyber-resilience measures. Otherwise, it might be required to shut-down the CPS to a safe state. Cyber-resilience addresses also a quick and reliable recovery after a successful attack. Also here, reliable and trustworthy information on the integrity status of CPS components is useful to support the automatic determination of recovery actions that are required and that can be performed responsibly considering the observed or potential manipulations on CPS components.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

REFERENCES

[1] R. Falk and S. Fries, "Attesting the Trustworthiness of a Credential Issuer", The Tenth International Conference on Cyber-Technologies and Cyber-Systems CYBER 2025, Lisbon Portugal, 28 September 2025, pp. 8-12, 2025. [Online]. Available from https://www.thinkmind.org/articles/cyber_2025_1_20_80058.pdf 2026.05.07

[2] D. Hardt, "The OAuth 2.0 Authorization Framework", IETF RFC 6749, October 2012. [Online]. Available from https://datatracker.ietf.org/doc/html/rfc6749 2026.05.07

[3] IEC 62443, "Industrial Automation and Control System Security" (formerly ISA99). [Online]. Available from: http://isa99.isa.org/Documents/Forms/AllItems.aspx 2026.05.07

[4] "Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act), Document 32024R2847, November 2024. [Online]. Available from: http://data.europa.eu/eli/reg/2024/2847/oj 2026.05.07

[5] "Directive 2014/53/EU of the European Parliament and of the Council of 16 April 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of radio equipment and repealing Directive 1999/5/EC Text with EEA relevance", 10/2023. [Online]. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32014L0053 2026.05.07

[6] A. Regenscheid, "Platform Firmware Resiliency Guidelines", NIST SP 800-193, May, 2018. [Online]. Available from: https://csrc.nist.gov/publications/detail/sp/800-193/final 2026.05.07

[7] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", IETF RFC 3467, November 2003. [Online]. Available from https://datatracker.ietf.org/doc/html/rfc3647 2026.01.13

[8] ITU-T X.509 ISO/IEC 9594-8:2020, Rec. ITU-T X.509 (2019), Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks. 2020 [Online]. Available from: https://www.itu.int/rec/T-REC-X.509-201910-I/en 2026.05.07

[9] TCG, "DICE Layering Architecture Specification". Online]. Available from: https://trustedcomputinggroup.org/resource/dice-layering-architecture/ 2026.05.07

[10] DMTF, "Security Protocols and Data Models (SPDM)", DSP0274 Version V1.3.1. [Online]. Available from: https://www.dmtf.org/sites/default/files/standards/documents/DSP0274_1.3.1.pdf 2026.05.07

[11] B. Laurie, E. Messeri, and R. Stradling, "Certificate Transparency Version 2.0", IETF RFC9162, December 2021. [Online]. Available from: https://datatracker.ietf.org/doc/rfc9162/ 2026.05.07

[12] H. Birkholz, D. Thaler, M. Richardson, N. Smith, and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC9334, December 2023. [Online]. Available from: https://datatracker.ietf.org/doc/rfc9334/ 2026.05.07

[13] M. Ounsworth, H. Tschofenig, H. Birkholz, M. Wiseman, and N. Smith, "Use of Remote Attestation with Certification Signing Requests", IETF Draft, March 2025. [Online]. Available from https://datatracker.ietf.org/doc/draft-ietf-lamps-csr-attestation/ 2026.05.07

[14] P. Cain, D. Pinkas, and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", IETF RFC3161, August 2001. [Online]. available from https://datatracker.ietf.org/doc/rfc3161 2026.05.07

[15] D. Cooper et al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", IETF RFC5280, May 2008. [Online]. available from https://datatracker.ietf.org/doc/rfc5280 2026.05.07

[16] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3", IETF RFC8446, August 2018. [Online]. available from https://datatracker.ietf.org/doc/rfc8446 2026.05.07

[17] IEEE 802.1x, "Port-Based Network Access Control", 2020. [Online]. Available from https://ieeexplore.ieee.org/document/9018454 2026.05.07

[18] R. Falk and S. Fries, "Enhanced Attack Resilience within Cyber Physical Systems", Journal on Advances in Security, vol. 16, no. 1&2, pp. 1-11, 2023. [Online]. Available from: https://www.iariajournals.org/security/sec_v16_n12_2023_paged.pdf 2026.05.07

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A Life-Cycle Model for Domain Security With Deterministic Multi-Criteria Relevance Scoring for Look-Alike Domains

1st Thomas Fritzler
Hochschule Darmstadt (h_da)-University of Applied Sciences
member of European University of Technology (EUt+)
Department of Computer Science
Darmstadt, Germany
thomas@fritzler.me

2nd Michael Massoth
Hochschule Darmstadt (h_da)-University of Applied Sciences
member of European University of Technology (EUt+)
Department of Computer Science
National Research Center for Applied Cybersecurity ATHENE
Darmstadt, Germany
michael.massoth@h-da.de

***Abstract*—This paper presents a phase-based security analysis of the domain-name life cycle - pre-registration, active registration, expiry, and malicious re-registration. Synthesizing peer-reviewed studies, documented incidents, and current threat intelligence (2014–2025), we map key attack vectors (for example typosquatting, dangling records, registrar compromise, expired-domain abuse) to concrete mitigations (registrar hardening, zone hygiene, renewal governance) and align them with representative control anchors in internationally recognized standards and regulatory requirements. This extended version substantially expands the original analysis by the following contributions: (i) deepening the standards alignment (including Network and Information Security Directive 2 implications), (ii) introducing a formal, deterministic multi-criteria relevance scoring model for look-alike domains, and (iii) outlining a concrete research agenda for empirical validation. The contribution remains a conceptual, literature-based synthesis; no new measurements are introduced. We argue that domain names are critical security assets that require continuous management across technical and administrative controls. Overall, the paper contributes a practical and standards-aligned framework that helps organizations, regulators, and researchers reason systematically about domain-related risks and prioritize mitigation efforts across the full domain name life cycle.**

Keywords*- *Domain Life Cycle; Domain Security; Domain Management; Look-alike Domains; Multi-Criteria Relevance Scoring.

I. Introduction

An earlier conference version of this work [1] introduced a phase-based security analysis of the domain-name life cycle and identified key threat patterns across registration, operation, expiry, and re-registration. The present paper substantially extends that work by deepening the standards alignment, formalizing a deterministic multi-criteria relevance scoring model for look-alike domains, and outlining a concrete research agenda for empirical validation.

Domain names form the backbone of navigation on the Internet. They function both as a company's **digital identity** and as **trusted anchors** for users accessing web resources [2]. A compromised domain can therefore trigger wide-ranging consequences - from phishing attacks to the complete takeover of online services. Attackers systematically exploit these weaknesses by operating with legitimate domains or deceptively similar names in order to bypass security mechanisms. In doing so, virtually every attack that relies on a seemingly legitimate sender or web address to evade defenses is facilitated [3].

Against this backdrop, the present paper analyzes the **vulnerabilities throughout the entire life cycle of a domain** - from registration, through operation and expiration, to potential takeover by third parties. The objective is to highlight *technical attack vectors* and *documented incidents* for each phase and to demonstrate their relevance for enterprises. To this end, existing scientific studies, security reports, and recorded attacks are comparatively evaluated. In addition, well-established tools are presented that can help to detect and prevent such weaknesses. **This contribution is a conceptual, literature-based synthesis rather than an empirical measurement study.** We address the following Research Questions (RQs): **RQ1** - What threats emerge at each phase of the domain name life cycle? **RQ2** - What technical and organizational controls effectively mitigate them? **RQ3** - How can the relevance of look-alike domains referencing a protected identifier be operationalized with a deterministic multi-criteria scoring algorithm? **RQ4** - What gaps suggest directions for future empirical validation?

a) Contributions of the Extended Version: Compared to the original conference version, this extended paper makes the following additional contributions:

- A formalized, deterministic relevance-scoring model for look-alike domains (Sections IV–V).
- An extended mapping of domain life-cycle threats to internationally recognized standards and regulatory requirements, including Network and Information Security Directive 2 (NIS2) implications (Section III and Table III).
- A structured discussion of open research challenges and directions for empirical validation (Section VI).

The remainder of this paper is structured as follows: **Section II** reviews related work and outlines our methodology; **Section III** presents the phase-oriented model of the domain name life cycle and a consistent, phase-by-phase threat mapping, including standards alignment; **Section IV** formalizes the deterministic multi-criteria relevance scoring model; **Section V** defines the scoring categories and criteria; **Section VI** discusses limitations and provides a research agenda for empirical

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

validation and practical adoption; **Section VII** concludes the paper.

II. Related Work

Prior studies typically focus on isolated threat surfaces. Typosquatting and other naming-confusion attacks have been analyzed in depth [4], [5], while the risks of dangling Domain Name System (DNS) records [6], [7] and expired-domain takeovers [8] have been explored separately. Adjibi *et al.* extend this line of research by quantifying how the Fortune 500 pursue *defensive registrations* and showing that roughly three-quarters of look-alike domains remain under third-party control [2]. Their work highlights a critical blind spot - corporate protection tends to begin *after* registration - and therefore does not capture threats that arise *before* or *after* that window (e.g., pre-registration brand monitoring or post-expiration abuse). **In contrast to prior work, this paper systematically maps threats across all four phases and couples them with phase-specific mitigations for practitioners.**

Other surveys take a broader perspective on DNS security. Schmid provides a view on systemic threats in DNS Insecurity [9], while Ramdas *et al.* catalog mitigation techniques against DNS-related attacks [10]. Affinito *et al.* examine domain lifetimes and quantify baseline risks across the ecosystem [11]. However, none of these works integrates security threats across the *entire domain life cycle*.

Methodology: We conducted a structured literature scan (2014-2025) across Association for Computing Machinery Digital Library (ACM DL), Institute of Electrical and Electronics Engineers (IEEE) Xplore, USENIX, Network and Distributed System Security Symposium (NDSS) and selected industry reports. Inclusion: peer-reviewed security work on DNS/domain life cycle, empirical incident reports; Exclusion: marketing/duplicate blog posts. Search keys included ”domain life cycle security”, ”dangling DNS”, ”expired domains”, ”typosquatting”, ”registrar hijacking”. Two reviewers screened titles/abstracts; we extracted threats, affected life cycle phase, and mitigation classes. This is a conceptual synthesis without new measurements.

Despite these valuable contributions, the literature still lacks an integrated framework that maps vulnerabilities from pre-registration to post-expiration. This paper closes that gap by proposing a phase-oriented security model and by deriving unified mitigation guidelines that link technical, administrative, and policy controls. Table I summarizes the comparative overview of existing reviews and how this work differs.

TABLE I
Comparative overview of existing reviews and how this work differs. Own synthesis based on the cited sources.

Work	Focus/Method	Life cycle coverage	Added value of this paper
COMST'21 [9]	Broad DNS threats survey	Cross-cutting; not phase-based	End-to-end, phase-based map (P1-P4) with aligned mitigations
ICCS'19 [10]	DNS attack mitigations (short survey)	General DNS; no life cycle lens	Life cycle view + concrete phase-wise controls
TMA'22 [11]	Domain lifetimes / expiry risks	Emphasis on lifetime/expiry	Integrates pre-reg, active, expiry, re-reg threats
NDSS'25 [2]	Defensive registrations (Fortune 500)	Narrow pre/post around reg.	Complements with ops/expiry abuse beyond registration

III. Domain Life Cycle and Its Vulnerabilities

The life cycle of a domain can be simplified into four phases: **(1) Pre-registration**, **(2) Active Registration**, **(3) Expired Domains**, and **(4) Malicious Re-registration**. While Phase 3 resembles Phase 1 in terms of availability, residual references (e.g., third-party logins, old references to email addresses, or inbound links) may still exist. Each phase poses distinct security threats [11]. *Per phase, we (i) define scope, (ii) enumerate threats, (iii) list recent evidence, and (iv) summarize mitigations.* Table II summarizes the key threats and defenses across the four phases addressed in this work. The following subsections outline these phases and their typical weaknesses.

A. Phase 1: Pre-registration

In the first phase, a domain is completely **freely available** - either never registered before or released by its previous owner - and can be registered anew. Even though no legitimate content exists under such a name, attackers can still **abuse** it by proactively registering it [12].

T1. Typosquatting and look-alike domains: Typosquatting refers to registering domain names that are **confusingly similar** to a well-known brand, usually through typographical errors or minor spelling changes. Users who mistype or fail to notice the difference are silently routed to the wrong site. This technique has deceived users since the early commercial Internet. Typical variants include letter transpositions, character omission or insertion, or using **homographs** - visually similar characters from other scripts. A well-known early case was `goggle.com`, intended to mislead visitors to `google.com`. Modern attackers increasingly rely on Internationalized Domain Names (IDNs) that mix, e.g., Cyrillic and Latin letters - an attack most browsers now detect and block [13].

Despite longstanding defensive efforts, typosquatting remains highly prevalent and continues to evolve. In 2024, for example, a security vendor reported a surge of Bifrost-Trojan campaigns leveraging VMware typosquats [14]. Fraudulent job sites and even parts of the SolarWinds supply-chain attack also traced back to typosquatting domains [15]. An Akamai analysis found that about 20.1% of all newly observed domains it tracks - roughly 13 million malicious domains every month, i.e., well over three million each week - are flagged as malicious, many of them look-alike registrations [16]. These are not accidental errors but **deliberate registrations by criminals**. The technique now underpins sophisticated fraud

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

schemes, such as combining bogus websites with matching social-media profiles, intercepting emails (e.g., Business-Email-Compromise), or smuggling trojanized code into development environments. One example is the discovery of Python libraries like `''requessts''` or `''reqquests''`, typosquats of popular packages registered on look-alike domains to trick developers [4].

a) Example: the Mastercard typo: A real-world incident underscoring the danger of small DNS errors is the *Mastercard typo*. A **faulty** DNS entry went unnoticed for years, meaning attackers could have registered the misspelled domain and weaponized it [17]. The case shows that even an innocuous typo in a DNS zone can open major security gaps, because customers or internal systems may unknowingly resolve the "wrong" domain.

b) Example: the BYD domain confusion: A recent real-world incident illustrates how brand visibility can backfire when obvious look-alike domains are left unprotected. During UEFA EURO 2024, Chinese electric-vehicle maker BYD bought prominent pitch-side advertising. Curious spectators naturally tried the German country code Top Level Domain (ccTLD) variant `byd.de` - but that name was already owned by an unrelated adult-toy retailer. The unexpected spotlight drove a massive traffic spike to the site, forcing its owners to publish a disclaimer that they were not affiliated with the car company [18]. The episode shows that even legitimate marketing can funnel large crowds to misleading domains, creating fertile ground for fraud or malware if criminals register similar names first. Careful domain-portfolio management across relevant top-level domains (TLDs) is therefore essential throughout the life cycle of a brand.

Scientific studies confirm typosquatting's breadth. A 2014 USENIX study analyzing hundreds of thousands of domains found that typosquatting is **widespread** and growing [5]. Actors invest significant resources to monetize these domains. More recent surveys reveal that **74% of look-alike domains** targeting Fortune 500 companies are held by third parties [2], highlighting aggressive coverage by criminals and domain speculators. Such names are often used for **phishing**, fraud, or brand abuse.

c) Standards mapping: Typosquatting and other look-alike registrations are rarely named explicitly in management-system standards; nevertheless, the underlying risk (external impersonation, credential theft, and malware delivery via confusable identifiers) is covered through requirements for risk-based control selection, external threat monitoring, and user-focused protections. Within ISO/IEC 27001/27002, preventive and detective coverage is primarily provided via threat intelligence and monitoring (Annex A: A.5.7, A.8.16), awareness and training against social engineering (A.6.3), and technical exposure reduction such as web-content controls and malware defenses (A.8.23, A.8.7), complemented by incident triage and response processes (A.5.25–A.5.27) [19], [20]. In Service Organization Control 2 (SOC 2) examinations, typosquatting-driven phishing aligns with the Trust Services Criteria for logical access safeguards and identity-related controls (CC6), as well as continuous monitoring and anomaly detection (CC7) and risk assessment / mitigation (CC3, CC9) [21]. For entities subject to the NIS2 Directive, proactive detection of impersonation domains and response playbooks operationalize the mandated cybersecurity risk-management measures (Art. 21(2), especially (a), (b), (f), (g), (j)); if exploitation results in a significant incident, the reporting regime of Art. 23 becomes relevant [22]. As additional widely used baselines, NIST CSF 2.0 (e.g., PR.AT, DE.CM/DE.AE, RS.*) and the CIS Controls v8.1 (notably Controls 9, 14, and 17) capture the same requirement class through awareness, continuous monitoring, and incident response management [23], [24].

While our phase-based model highlights typosquatting and look-alike domains as a persistent and highly practical threat vector, current defensive practice is still largely binary: a domain is either defensively registered or left to third parties. In reality, organizations face thousands of potential permutations per brand and must decide the names to register, monitor, or ignore. This raises an open question that is not answered by existing work: how can we prioritize look-alike domains in a principled, risk-based way?

To address this gap, the extended version of this paper introduces a formal, deterministic relevance-scoring model for look-alike domains (Sections IV–V). The goal is to map look-alike candidates into a small set of actionable categories (e.g., "crown jewels", "critical", "important", "low priority", "irrelevant") and to link each category to concrete mitigation strategies such as defensive registration, continuous monitoring, or deliberate non-action. The score is designed as an upstream triage signal rather than a claim of maliciousness; Section VI outlines open research challenges and a concrete agenda for empirical validation and calibration.

T2. Orphaned and published domains: A threat that spans **Phase 1 (Pre-registration)**, also surfaces during **Phase 2 (Active Registration)** through abandoned or misconfigured *subdomains*, and re-appears in **Phase 4 (Malicious Re-registration)**. Attackers actively look for **domain or subdomain names that are still referenced** - in documentation, configuration files, code snippets, or lingering DNS records - yet are either unclaimed or left dangling. Although these names look legitimate from the outside, they are **not actually under the rightful owner's control**. By claiming an orphaned domain or subdomain, adversaries can invisibly insert themselves into traffic intended for the original destination [6], [25], [26]. Modern scanners such as *BadDNS* now crawl websites to locate externally referenced, takeover-able domains [27].

d) Standards mapping: Orphaned or externally referenced-but-uncontrolled (sub)domains constitute an asset-governance and decommissioning failure that standards typically address indirectly via asset management, configuration control, and secure change processes rather than by naming the attack. In ISO/IEC 27001/27002 terms, the core requirement is to maintain an accurate inventory of information-related assets and associated identifiers (A.5.9) and to enforce controlled configuration and lifecycle changes (A.8.9, A.8.32), complemented by monitoring for externally

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

triggered security events (A.8.16) and documented operational procedures for deprovisioning and cleanup (A.5.37) [19], [20]. SOC 2 addresses the same class of failure through criteria covering change management and configuration discipline (CC8) and monitoring for deviations and security events (CC7), with the governance/risk layer (CC3/CC9) providing the rationale for systematic retirement checklists and periodic scans [21]. Under NIS2, orphaned-domain prevention and detection can be positioned as part of the required risk-analysis and security policies (Art. 21(2)(a)), security in system acquisition/maintenance including vulnerability handling (Art. 21(2)(e)), effectiveness testing (Art. 21(2)(f)), and asset/access governance (Art. 21(2)(i)) [22]. Complementary baselines such as NIST CSF 2.0 (ID.AM, PR.*) and CIS Controls v8.1 (Controls 1, 2, and 4) make the same expectation explicit: unmanaged identifiers are unmanaged attack surfaces [23], [24].

B. Phase 2: Active Registration

In Phase 2 the domain is under **active ownership** (typically by an organization) and used for services such as websites, email, or Application Programming Interface (APIs). Although the name is legitimately controlled, numerous **attack vectors arise from misconfiguration or insufficient safeguards**. The main concerns are **DNS configuration**, registrar security, and subdomain management [3].

T1. DNS Misconfigurations and Gaps: A domain is only as secure as its DNS settings. Faulty or negligent configuration can give adversaries opportunities to abuse or manipulate the name. Key issues include:

Missing DNSSEC signing: Domain Name System Security Extensions (DNSSEC) cryptographically signs DNS responses to guarantee their **authenticity and integrity**. Without DNSSEC, domains are vulnerable to cache-poisoning and manipulation - an attacker could inject forged answers and redirect users to malicious Internet Protocol (IP) addresses. Despite clear benefits, adoption remains low: an Asia-Pacific Network Information Centre (APNIC) study in 2023 found that **only about 4.3 % of .com domains are DNSSEC-signed** [28]. In other words, more than 95 % lack this protection. Some TLDs fare better - `.nl` reaches roughly 60 % coverage [28] - yet a global gap persists. Technical complexity, operational effort, and limited know-how leave many domains exposed whenever an attacker can influence a resolver or intercept traffic.

Missing CAA records: A **Certification Authority Authorization (CAA)** record lets owners specify the Certification Authorities (CAs) that may issue Transport Layer Security (TLS) certificates for the domain. Without CAA, any CA could issue a certificate (assuming domain-control checks can be bypassed via DNS tampering or error). CAA therefore reduces the risk of *unintended or fraudulent issuance*. Nevertheless, a 2020 survey showed that **only about 3 % of the Alexa Top-1-Million** domains publish a CAA record. Given the ease of deployment, this figure is strikingly low. A CAA record might restrict issuance to `Let's Encrypt` or `DigiCert`; some domains even use `issue ';'` to permit **no** CA - 358 cases in that study [29].

Open zone transfers (AXFR): Zone transfers replicate data between authoritative name servers. If a server is misconfigured to allow AXFR from *unauthorized IPs*, attackers can pull a **complete zone dump**. The dump reveals all records - internal subdomains, IP mappings, etc. - and aids further attacks. Although Common Vulnerabilities and Exposures entry CVE-1999-0532 highlights the risk, Internet scans as late as 2016 still found "large numbers" of exposed servers. Successful leaks expose hidden services such as `vpn.company.com` or `dev.db.company.com`. Mitigation is trivial: allow transfers **only to authorized secondary servers** [30].

Stale or incorrect DNS records: Domains or subdomains often move or services are retired without cleaning all records. Such **stale entries** may point nowhere - or worse, be taken over (see *subdomain takeover*). A special case is **incorrect name-server (NS) entries**: if registry-level name servers are misspelled or unresponsive, the domain becomes unstable. Attackers have exploited these situations. The 2024 *"Sitting Ducks"* campaign revealed hundreds of thousands of domains with **DNS misconfigurations** (e.g., bad NS pointers) effectively abandoned [31]. Attackers impersonated the intended name servers and seized control [31]. Infoblox found nearly **800,000 vulnerable domains** in three months; about 9 % (>70,000) were **actively hijacked** [31]. These domains - often legitimate but misconfigured - were then abused for phishing, investment fraud, and more [31]. Even a minor lapse (e.g., an outdated NS entry) can turn a domain into *easy prey*.

A common scenario is a DNS record that points to an external domain (e.g., via **canonical name (CNAME)**, **mail exchanger (MX)**, or **name-server (NS)** records) no longer controlled by the organization. Such a **dangling record** invites abuse: registering the missing domain diverts traffic to the attacker. For instance, `example.com` might include `oldservice.example.com CNAME oldservice-provider.com`, even though `oldservice-provider.com` no longer exists. Whoever registers that domain gains control over every request to `oldservice.example.com`. The same applies to unregistered domains listed as **MX** or **NS**, enabling email interception or name-server takeover [6], [26].

In short, during Phase 2 the owner must ensure that all security-relevant DNS settings are correct and current. Misconfiguration directly threatens the **integrity of name resolution**. Additional precautions include avoiding unnecessary internal disclosures (e.g., chatty TXT records or revealing subdomain names) and routinely auditing the zone for **anomalous entries**.

a) Standards mapping: DNS misconfiguration risks (e.g., missing DNSSEC/CAA, permissive zone transfers, stale delegation and dangling records) are addressed in most standards as a configuration-management and secure-network-services problem, with DNSSEC/CAA additionally grounded in technical Internet standards. ISO/IEC 27001/27002 cover these controls primarily through configuration management and controlled change (A.8.9, A.8.32), network security and secure

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

network services (A.8.20–A.8.21), continuous monitoring and logging (A.8.15–A.8.16), and cryptographic controls where authenticity/integrity protection is required (A.8.24), which is directly applicable to DNSSEC deployment [19], [20]. SOC 2 similarly emphasizes controlled change (CC8) and monitoring for misconfiguration and anomalous system behavior (CC7), with risk assessment (CC3) motivating DNS baselines and periodic audits [21]. For NIS2-regulated entities, systematic DNS hardening and audits operationalize Art. 21(2)(a) (risk analysis and system security), Art. 21(2)(e) (secure acquisition/development/maintenance incl. vulnerability handling), Art. 21(2)(f) (effectiveness assessment), and Art. 21(2)(h) (cryptography/encryption policies) [22]. At the technical standard level, DNSSEC is specified in RFC 4033–4035 and CAA in RFC 8659; referencing these RFCs improves methodological transparency by distinguishing normative protocol requirements from organizational control objectives [32]–[35].

T2. Threats from subdomain takeover: Many organizations delegate subdomains to external cloud providers - e.g., `shop.example.com` for Software as a Service (SaaS) or `cdn.example.com` for a Content Delivery Network (CDN) - via **CNAME** records (such as `shop.example.com CNAME shopsaas.com`). While the service is active and configured, no issue arises. **The threat arises when external services are decommissioned but DNS entries remain.** The subdomain then points to a *non-existent* target - a *dangling DNS record*. Attackers can **re-register** or claim that resource (e.g., the freed **cloud host name or account**) to seize control [6], [26]. This is known as a **subdomain takeover**. The attacker "captures" a victim's subdomain by obtaining the referenced external domain or resource. Once in control, they can serve content or intercept traffic under the trusted hostname. Cloud platforms are frequent targets: Azure Web Apps, Amazon Web Services (AWS) S3 buckets, GitHub Pages, Heroku, and more. If the CNAME is left behind after deletion, an attacker can spin up an identically named service and **take over** the subdomain [6], [26].

Empirical work shows subdomain takeover is not rare but a **systemic risk**: in 2016 researchers identified 467 vulnerable subdomains among the Alexa Top-10k plus university sites [7]. Follow-up scans across cloud platforms found over **700 000 vulnerable DNS entries** overall [6], [8].

Root causes are usually poor cleanup: projects end, services migrate, yet DNS records linger. Especially in DevOps and cloud cultures where teams create subdomains autonomously, visibility is lost. Companies should schedule **regular audits** of their DNS zones to detect *dangling* entries. Tools such as **Subjack**, **Subzy**, takeover modules in scanners like **Nuclei**, or the comprehensive **BadDNS** can automate detection by checking CNAME/MX/NS targets for registrability [27].

b) Standards mapping: Subdomain takeover is primarily a lifecycle and supplier-integration risk: delegated cloud resources are retired or reassigned while DNS pointers persist, enabling adversaries to rebind trusted hostnames to attacker-controlled infrastructure. ISO/IEC 27001/27002 treat this as a combination of cloud-service governance and supplier control (A.5.23, A.5.21–A.5.22), configuration/change discipline (A.8.9, A.8.32), and monitoring for externally observable security events (A.8.16), with incident assessment/response processes (A.5.25–A.5.27) providing corrective capability once takeovers are detected [19], [20]. SOC 2 examinations typically map such risks to change management (CC8), monitoring/system operations (CC7), and risk mitigation (CC9), especially when third-party services host customer-facing endpoints or authentication flows [21]. NIS2 explicitly elevates supply-chain and service-provider security to a minimum measure (Art. 21(2)(d)); combined with Art. 21(2)(e) and (f), this provides a direct normative basis for periodic scans of delegated DNS targets, provider offboarding controls, and documented verification of cleanup effectiveness [22]. As further baselines, NIST CSF 2.0 (notably GV.SC and DE.CM) and CIS Controls v8.1 (Controls 4, 12, 15, and 16) support the same requirement class: unmanaged external dependencies become attacker-controlled trust boundaries [23], [24].

T3. Weaknesses in Registrar Security: Domain protection also depends heavily on the **security of the registrar account** - the portal where the name is registered and managed. An attacker who gains access can **seize the entire domain**: redirect name servers, change ownership details, or transfer the domain to another account.

Numerous incidents of **domain hijacking** stem from social engineering or registrar breaches. Attackers exploit weak passwords, absent two-factor authentication, or technical flaws at smaller registrars. A prominent example is the **"Sea Turtle"** campaign (2017-2019), during that campaign, a state-sponsored actor compromised registrars and DNS providers to hijack high-profile domains - mainly in the Middle East - by phishing credentials and altering DNS to their own servers, even acquiring valid TLS certificates [36]. This illustrates that even perfectly configured DNS zones fail if **registrar infrastructure is breached**.

Cyber-criminals without state backing also hijack domains. Often they first compromise the owner's email address (e.g., by registering an expired domain from Phase 3), then trigger a **password reset**. Support staff may also be fooled into **unauthorized transfers**. High-value domains - famous brands or premium .com generics - regularly appear in news reports after such thefts [36]. In 2015, for instance, Lenovo's domain was redirected to a defacement server, allegedly via registrar account compromise [37].

Key **weak points** include absent **multi-factor authentication**, lack of **Registry Lock** (a service that blocks critical changes unless manually approved), and poor credential hygiene. Registry Lock is highly effective yet mainly adopted by large firms: **46 % of companies using enterprise registrars** employ it, versus only 7 % using mass-market registrars. Many Small and Medium-sized Enterprises (SMEs) may not know or purchase the feature, and some registrars do not actively promote it [38].

Altogether, Phase 2 requires a **holistic defense** of the domain: both technical DNS parameters and administrative access must be secured, or attackers may gain total control

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

- with potentially catastrophic outcomes such as fraud, reputation damage, or data breaches.

c) Standards mapping: Registrar compromise and domain hijacking represent a high-impact identity and access management failure at the authoritative control plane of a domain, with downstream effects on integrity of name resolution, email authenticity, and service trust anchors. ISO/IEC 27001/27002 address this risk through identity and access governance (A.5.15–A.5.18), privileged access control (A.8.2), secure authentication (A.8.5), and auditability via logging and monitoring (A.8.15–A.8.16), complemented by structured incident handling and evidence preservation (A.5.24–A.5.28) [19], [20]. SOC 2 maps registrar-account protection to the logical access criteria (CC6, including strong authentication and protection of credentials), to monitoring/system operations (CC7), and to risk mitigation (CC9), because registrar compromise is often realized through phishing/social engineering and weak recovery processes [21]. Under NIS2, strong authentication and access governance are explicitly required as minimum measures (Art. 21(2)(i)–(j)); given the potentially significant impact of hijacking, Art. 23 reporting may be triggered when the event materially affects service provision or causes downstream compromise [22]. Further normative guidance on authentication assurance (e.g., NIST SP 800-63-4) and control baselines such as CIS Controls v8.1 (Controls 5–6, 8, and 17) provide additional, widely accepted operationalization for multi-factor authentication (MFA), recovery hardening, and continuous monitoring [24], [39].

C. Phase 3: Expired Domains

Domains are typically registered for periods of 1 to 2 years and must be renewed regularly. If a domain is **not renewed in time**, it enters an expiry workflow. The registrar first places it in a short **grace period** (typically 30-45 days, during this period, the original holder can recover it for a fee), optionally followed by a **redemption period** (another $\sim$30 days, usually with a higher restoration fee). Finally, the domain is deleted and released for **re-registration** [11]. From a corporate perspective, allowing a domain to lapse is usually unintentional, yet still occurs frequently, whether through organizational errors (missed reminder emails, staff changes) or because the name is deemed no longer important (e.g., after rebranding or project shutdown). Older corporate domains or those of acquired subsidiaries are especially prone to "slipping through" [40]. Once a domain becomes available, attackers have an excellent opportunity to register and exploit it. Phase 3 therefore shows the threats posed by **expired domains**.

T1. Abuse of expired domains for phishing and fraud: A **released domain** can be registered by anyone - professional domain traders (*drop-catchers*) often run automated scripts to acquire attractive names the moment they drop. Criminal actors monitor such drop lists for **promising targets**. Domains previously owned by well-known organizations are highly valuable because they **inspire trust**. Attackers re-register them and deploy **convincing replicas** of the original content to deceive users. A typical pattern is launching a **fake webshop** on a formerly legitimate domain [41]. Brian Krebs [42] chronicled how a photographer's lapsed portfolio domain was re-registered by fraudsters and converted into a counterfeit sneaker store. Visitors seeking her work unknowingly entered card details, which criminals harvested and resold. Besides reputational damage, the photographer lost access to linked accounts because attackers also seized her former email address [41]. Automated renewal management and systematic SaaS deprovisioning ensure that critical domains never lapse and that dormant integrations are fully removed after project shutdowns.

a) Standards mapping: Expired-domain abuse is primarily a lifecycle governance failure with externalized impact: once a trusted identifier lapses, it can be re-registered and used to exploit residual trust relationships with customers and partners. ISO/IEC 27001/27002 provide coverage through asset/identifier governance (A.5.9), operational procedures and responsibilities for controlled retirement (A.5.37, A.6.5), continuous threat intelligence and monitoring (A.5.7, A.8.16), and incident response capabilities when fraudulent use is detected (A.5.25–A.5.27) [19], [20]. SOC 2 similarly frames the risk through monitoring and incident response (CC7) and risk assessment (CC3/CC9), because renewal governance is a control activity that prevents externally exploitable conditions [21]. In NIS2 contexts, systematic renewal controls and watchlist-driven detection contribute to Art. 21(2)(a) and (b) (risk analysis and incident handling), Art. 21(2)(f) (effectiveness assessment), and Art. 21(2)(g) (cyber hygiene/training); significant exploitation may activate Art. 23 reporting timelines [22]. NIST CSF 2.0 and CIS Controls v8.1 provide compatible baselines by tying asset governance and monitoring to detection/response outcomes (e.g., DE.CM and Controls 1/17) [23], [24].

T2. Email and account takeovers: If a company abandons a domain once used for email addresses (e.g., `@oldcorp.com`), an attacker who re-registers it can intercept all future mail. They can impersonate the firm, send **phishing mails from the genuine domain**, or reset passwords of existing accounts [43]. Many online services rely on email for password recovery. If a former employee signed up to a cloud service with `name@oldcorp.com`, the new domain owner can use "forgot password" to gain access. Researchers warn that **trade secrets can leak** or attackers may penetrate personal accounts of ex-staff [43]. A 2025 report showed that registering domains of defunct start-ups yielded access to countless SaaS accounts. Examples included ChatGPT, Slack, Notion, Zoom, and even HR systems still tied to old email addresses [44]. Attackers viewed sensitive data such as tax forms, payslips, and applicant information [44]. The study exposed a *design problem* in Single Sign-On (SSO): many providers identify users solely by email domain (the `hd` claim in Google OAuth), so a fresh Google account under the captured domain is treated as the original user [44]. Even without historical mail access, simply re-creating the address **preserves perceived identity** and unlocks accounts [44].

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

b) Standards mapping: Email- and account-takeover risks after domain expiry combine authentication weaknesses (recovery flows bound to email domains) with residual identity assumptions in SSO ecosystems. ISO/IEC 27001/27002 map these risks to secure authentication and credential governance (A.8.5; A.5.16–A.5.17), access rights management (A.5.18), logging and monitoring for anomalous account activity (A.8.15–A.8.16), and incident handling processes to contain compromised identities (A.5.25–A.5.27) [19], [20]. SOC 2 addresses the same risk class via the logical access criteria (CC6, including credential protection and strong authentication) and via monitoring/incident response (CC7), because takeover is often initiated through phishing or recovery abuse [21]. NIS2 explicitly requires MFA or continuous authentication solutions and secure communications where appropriate (Art. 21(2)(j)), together with access control and asset management (Art. 21(2)(i)) and incident handling (Art. 21(2)(b)); material takeovers may trigger Art. 23 reporting [22]. Additional normative guidance on digital identity assurance (NIST SP 800-63-4) and incident handling practice (e.g., NIST incident response guidance) complements CIS Controls v8.1 controls for account/access management and incident response (Controls 5–6 and 17) [24], [39].

T3. Threats from lingering integrations: Beyond email, an expired domain may still be embedded in security infrastructure. Examples include **OAuth redirect URIs** or API callbacks. If such URLs point to a domain later relinquished, an attacker who captures it can control the OAuth flow. One case involved an integration using a subdomain of a now-defunct firm as its OAuth redirect; [44] after researchers bought the domain, they could masquerade as an authenticated organization and access data [44]. Likewise, **Single Sign On (SSO) endpoints**, API keys, or license servers may rely on the old domain. Re-using former names thus creates **bridges into previously protected areas** that the victim no longer monitors [43], [44].

c) Standards mapping: Lingering integrations (OAuth redirects, SSO endpoints, API callbacks) are a composite of supply-chain and change-management risks: trust is delegated to identifiers that may later become attacker-controlled. ISO/IEC 27001/27002 address this through supplier and cloud governance (A.5.21–A.5.23), controlled change and configuration management (A.8.9, A.8.32), secure authentication and access governance for integration identities (A.5.16–A.5.18; A.8.5), and monitoring/incident response once anomalous callbacks or redirect abuse occurs (A.8.16; A.5.25–A.5.27) [19], [20]. SOC 2 maps these issues to change management (CC8), logical access controls around integrations (CC6), and monitoring/system operations (CC7), with risk mitigation criteria (CC9) supporting third-party integration inventories and periodic validation of redirect/callback allowlists [21]. NIS2 provides a direct normative anchor via Art. 21(2)(d) supply-chain security and Art. 21(2)(e) secure maintenance, combined with effectiveness assessment (Art. 21(2)(f)) and access/asset governance (Art. 21(2)(i)–(j)) [22]. NIST CSF 2.0 (GV.SC; PR.*) and CIS Controls v8.1 (Controls 15–16 and 17) supply compatible operational baselines for third-party integration governance and incident handling [23], [24].

T4. Drop-catching and domain speculation: Drop-catching - automated registration of recently expired names - is not inherently criminal; an entire legitimate industry resells such domains. Fraudsters, however, also exploit it as a **business model**. With little effort, they obtain domains that already **carry trust** (user familiarity, positive mail reputation, inbound links) [41]. These names are then *monetized*: directly for phishing or fraud, resold to the original owner (cybersquatting/extortion), or used for **spam/SEO** [45]. Studies show most seized domains end up in **black-hat SEO** networks - injecting links and redirects to boost dubious sites [45], [46].

A headline example illustrating speculative risk occurred in 2021: Google's official Argentinian domain, `google.com.ar`, briefly lapsed and was bought for about $3 USD by a local web designer [47]. No harm ensued - he promptly returned it - but the incident shows even a tech giant can stumble, and the potential impact had an attacker acted maliciously [47].

d) Standards mapping: Drop-catching and speculative acquisition of recently expired domains is not, in itself, a control objective in major standards; however, the resulting risk (loss of control over organizational identifiers and residual trust) is covered via risk management, asset governance, and legal/contractual protection mechanisms. ISO/IEC 27001/27002 provide relevant coverage through threat intelligence and external monitoring (A.5.7), asset inventory and lifecycle accountability for identifiers (A.5.9; A.5.37), and compliance/legal controls that frame defensive strategies such as dispute processes or contractual registrar protections (A.5.31–A.5.32) [19], [20]. SOC 2 treats the domain-expiry attack surface as a risk assessment and mitigation topic (CC3/CC9), supported by monitoring (CC7) for signals of third-party acquisition and subsequent abuse [21]. In NIS2 contexts, preventative renewal governance and monitoring support Art. 21(2)(a) and (f) (risk analysis and effectiveness assessment); moreover, for entities providing domain name registration services, NIS2 introduces specific obligations on registration data accuracy and verification (Art. 28), which can improve traceability and reduce abuse at ecosystem level [22]. As additional baselines, NIST CSF 2.0 emphasizes identifier-related risk under governance and risk assessment categories, while CIS Controls v8.1 situate the mitigation in asset management and incident response controls (Controls 1 and 17) [23], [24].

Phase 3 therefore focuses on **keeping expired domains out of adversaries' hands**. As shown, both external users (customers) and the organization itself (through account takeover) can fall victim otherwise. For attackers, orphaned domains are attractive **attack platforms**, cloaked in legitimacy and leveraging existing trust relationships.

Phase boundary clarification: Phase 3 describes the loss of organizational control over a domain identifier (typically through non-renewal and release into the public registration pool), whereas Phase 4 focuses on adversarial control and

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

monetization *after* re-registration or takeover.

D. Phase 4: Malicious Re-registration

Phase 4 examines the situation after a domain has been taken over by a *new owner* - often an attacker or speculator. The central question is: **How can a captured domain be exploited or monetized?**

T1. Threats from domain speculation and trading: Some actors register domains merely to resell them at a profit. In the harmless case, these are generic terms (e.g., `domaintrading.net`) or catchy names awaiting a buyer. It becomes problematic when speculators register domains clearly linked to a company or product and then try to sell them back to the rightful owner - often at inflated prices. This practice is known as **cybersquatting**. Victims may feel extorted into repurchasing the domain to prevent abuse or protect their brand. Immediately after expiration, fraudsters can seize a domain and demand a ransom for its return [2].

Large-scale domain parking is another issue: studies show that a significant share of *re-registered* or never-used domains are **parked** - they host no original content, only ad links, redirects, or placeholders. According to [45], major parking providers control hundreds of thousands of such domains. Besides *trademark concerns*, parking poses *security threats*: parked domains can be repurposed for phishing or malware without notice, and it is hard for outsiders to judge whether a parked domain is benign or simply "waiting" for abuse [45].

a) Standards mapping: Cybersquatting and large-scale domain parking are typically framed in standards as a combination of legal/identifier protection and security monitoring: the same third-party control of confusable identifiers can become a delivery vehicle for malware, phishing, or reputational harm. ISO/IEC 27001/27002 map the topic through legal and intellectual-property compliance requirements (A.5.31–A.5.32), threat intelligence and external monitoring (A.5.7; A.8.16), and exposure-reduction controls such as web filtering and malware defenses when parked domains are used as attack infrastructure (A.8.23; A.8.7) [19], [20]. SOC 2 similarly positions the risk under risk assessment/mitigation (CC3/CC9) and monitoring (CC7), particularly when parked domains are used to target customers or to redirect residual traffic from formerly legitimate identifiers [21]. Under NIS2, the same operational posture is justified through Art. 21(2)(a) and (b) (risk analysis and incident handling) and (f) (effectiveness assessment); additionally, for domain registration service providers, Art. 28 introduces measures that may reduce abusive registrations and increase accountability [22]. NIST CSF 2.0 and CIS Controls v8.1 complement this framing by making explicit that brand-/identifier-related monitoring is a legitimate input to detection and response programs (DE.CM; Controls 9/17) [23], [24].

T2. Criminal monetization options: Once attackers gain control of a domain - whether through typosquatting, expiration, or takeover - they acquire a highly valuable asset. **Possible revenue streams** include:

Phishing and fraudulent sites: As in Phase 3, a reputable domain can host convincing phishing pages. Login portals or data-theft forms seem trustworthy because the URL looks legitimate. **Fake shops** are equally common [41], [48].

Malware delivery and command-and-control: Attackers distribute malicious files or run **Command-and-Control (C2) infrastructure** for botnets and trojans on the domain. A name absent from blacklists enjoys better initial reach. Some *Sitting Ducks* domains were used for Traffic-Distribution Systems (TDS), spam, and **malware C2** servers [31], [41].

Spam and email scams: A freshly registered (or hijacked) domain can host mail servers to send spam. A legacy corporate domain may still have a **good sender reputation**, reducing filter hits [2]. Crime groups deliberately set up mail on lapsed domains and even used them to seize social-media or SaaS accounts, as noted earlier [41].

Leveraging residual integrations: If the original organization still references the domain in webhooks, OAuth redirects, or API keys, the new owner can exploit that linkage. Documented cases show attackers gaining **access to enterprise data** by posing as legitimate endpoints [41], [44].

Redirects and traffic parking: A subtler yet harmful tactic is funneling all residual traffic to ad or affiliate sites. High-traffic domains earn click revenue via automated parking platforms. While not always illegal, such redirects can tarnish brand reputation - e.g., a former corporate site suddenly points to gambling or adult content [41], [45].

b) Standards mapping: Criminal monetization of seized domains (phishing, malware delivery, command-and-control, and spam) corresponds to the core incident-prevention and incident-response objectives of major standards and is therefore among the most directly covered threat classes. ISO/IEC 27001/27002 provide a comprehensive control basis via anti-malware and network protections (A.8.7; A.8.20–A.8.21), logging and monitoring (A.8.15–A.8.16), technical vulnerability management (A.8.8), and a structured incident management lifecycle from preparation and triage to response and learning (A.5.24–A.5.27) [19], [20]. SOC 2 aligns this activity with the monitoring/system operations criteria (CC7), logical access protection (CC6), and incident response/risk mitigation expectations (CC9), as required to detect and contain malicious infrastructure hosted under an organization-associated identifier [21]. For NIS2 entities, these controls map directly to the minimum cybersecurity risk-management measures, notably incident handling and business continuity (Art. 21(2)(b)–(c)), effectiveness assessment (Art. 21(2)(f)), and secure communications/authentication (Art. 21(2)(j)); exploitation that materially affects services typically intersects with Art. 23 reporting obligations [22]. Complementary baselines such as NIST CSF 2.0 (DE.*, RS.*, RC.*) and CIS Controls v8.1 (Controls 8, 10, 13, and 17) provide widely used operationalizations for logging, malware defenses, network monitoring, and incident response programs [23], [24].

Overall, Phase 4 demonstrates that **attackers have many ways to profit from a seized domain**, whether financially or within larger campaigns. A domain in hostile hands becomes a **"weapon"** that slips past defenses thanks to its trusted name. For enterprises, **preventing** such takeovers must be top

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE II
SYSTEMATIC THREAT MAPPING ACROSS DOMAIN LIFE CYCLE PHASES. OWN SYNTHESIS BASED ON THE CITED SOURCES.

Phase	Threat (ID)	Recent evidence	Primary controls
P1	T1 Typosquatting / look-alikes	USENIX'14 [5]; NDSS'25 [2]	Defensive regs; brand watch
P1	T2 Orphaned/published names	NDSS'23 [25]; Unit 42 [26]	Monitoring; claims; cleanup
P2	T1 DNS misconfig	CISA [30]; Infoblox [31]	DNSSEC; CAA; restrict AXFR; audits
P2	T2 Subdomain takeover (dangling)	CCS'16 [7]; NSDI'24 [6]	Quarterly scans [27]; cleanup
P2	T3 Registrar account compromise	IMC'22 [36]; The Guardian [37]	MFA; Registry Lock; two-party recovery
P3	T1 Phishing / fraud on expired domains	S&P'22 [41]; KrebsOnSecurity [42]	Auto-renew; renewal governance; brand watch
P3	T2 Email / account takeovers	IMC'24 [43]; The Hacker News [44]	Deprovisioning; retirement checklist
P3	T3 Lingering integrations (OAuth / SSO / API)	IMC'24 [43]; The Hacker News [44]	Deprovisioning; OAuth/SSO audit; key revocation
P3	T4 Drop-catching / speculation	TMA'22 [45]; BBC News [47]	Backorder; legal (UDRP); retain redirects
P4	T1 Speculation/parking abuse	TMA'22 [45]; WIPO [46]	Monitoring; legal recourse
P4	T2 Criminal monetization (phishing / C2 / spam)	Infoblox [31]; SRLabs [48]	Takedowns; blocklists; incident response

priority, because **damage control** afterwards is costly. Once criminals register the domain, recourse is often limited to lengthy legal action (e.g., a Uniform Domain-Name Dispute-Resolution Policy (UDRP) proceeding) or an expensive settlement - long after harm may already be done.

IV. MULTI-CRITERIA SCORING ALGORITHM

The preceding chapters established the motivation for risk-based prioritization of domain registrations and for continuous monitoring of identifier-referencing domains within cyber threat intelligence workflows. Building on these requirements, the following section formalizes a deterministic 0–100 relevance scoring model and specifies its assumptions, inputs, and aggregation logic. We deliberately position the proposed scoring as a deterministic decision-support heuristic, not as a predictive classifier or a machine-learning model. This section thereby addresses **RQ3** by specifying a transparent, reproducible relevance metric that supports defensive registration, monitoring, and triage decisions for look-alike domains.

This section specifies a reproducible multi-criteria scoring function that maps a candidate domain name d to an integer-valued relevance score $\mathrm{Score}(d, B) \in \{0, \ldots, 100\}$ with respect to a protected identifier object B. The object B generalizes private-sector brands and trademarks as well as public-sector designations (e.g., names of authorities, ministries, agencies, programs, or sovereign digital identities). The score operationalizes *domain relevance* as whether and how strongly a domain name is (i) lexically or visually confusable with B, (ii) semantically suggestive of services, stakeholders, or jurisdictions associated with B, and therefore (iii) worthy of prioritized registration, monitoring, and defensive handling. In trademark contexts, these dimensions approximate factors that contribute to likelihood-of-confusion assessments; in public-sector settings, they quantify the propensity of a domain to be perceived as an official authority or service and to facilitate impersonation.

The score intentionally quantifies *relevance* rather than *maliciousness*. Domains used for benign purposes—including authorized resellers, press portals, criticism sites, or fan pages—can legitimately obtain high scores if they strongly refer to B. Consequently, a high score is interpreted as "monitoring priority" and not as a standalone justification for enforcement or takedown. Operationally, the score is designed as an upstream filter that reduces the candidate space for downstream analyses (e.g., content inspection, reputation signals, WHOIS/DNS telemetry, and legal assessment).

Given the integer-valued score $\mathrm{Score}(d, B)$ on a 0–100 scale, numeric values can be mapped to analytically defined risk bands in order to support prioritization and operational decision-making. Thresholds depend on the operating context (e.g., registry-scale monitoring vs. incident-driven triage) and should be calibrated against organizational objectives and analyst capacity. The following mapping is illustrative:

- $\mathrm{Score}(d, B) \geq 90$: *crown jewel* or highest-priority domain (strong association with B)
- $80 \leq \mathrm{Score}(d, B) < 90$: *critical look-alike*
- $60 \leq \mathrm{Score}(d, B) < 80$: *important look-alike*
- $30 \leq \mathrm{Score}(d, B) < 60$: *marginal look-alike*
- $\mathrm{Score}(d, B) < 30$: *irrelevant* or low-priority domain

These thresholds are included to make the mapping from a continuous ranking to actionable classes explicit; they can be calibrated on empirical data or adjusted to reflect an organization's risk tolerance.

Notation and inputs: The model evaluates the *registrable* domain (effective TLD + 1) in order to support decisions on registration and monitoring at the domain-registration level. Let $\mathrm{reg}(d)$ denote the registrable domain of a fully qualified domain name d, computed using the Public Suffix List. Write

$$\mathrm{reg}(d) = \ell(d) \,.\, \tau(d), \tag{1}$$

where $\ell(d)$ is the registrable *label* (i.e., the part without the TLD) and $\tau(d)$ is the (effective) TLD. Unless stated otherwise, criteria C_1–C_4 and C_6–C_9 are computed *only* over $\ell(d)$; criterion C_5 is the only criterion that explicitly depends on $\tau(d)$ or cross-TLD variants.

Let $\mathcal{B}_B$ denote a set of protected identifier strings associated with B, including the canonical name, abbreviations, transliterations, and sanctioned variants (e.g., "ExampleBank", "Example Bank", "EBank"). For public-sector objects, $\mathcal{B}_B$ analogously contains official designations and acronyms (e.g., "National Tax Authority", "NTA").

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE III
EXTENSION OF TABLE II: MAPPING DOMAIN-LIFE-CYCLE THREATS TO SELECTED INTERNATIONALLY RECOGNIZED STANDARDS AND NORMS. ISO MAPPINGS REFER TO ISO/IEC 27001:2022 ANNEX A (ALIGNED WITH ISO/IEC 27002:2022); SOC 2 MAPPINGS REFER TO THE AICPA TRUST SERVICES CRITERIA (TSP SECTION 100); NIS2 MAPPINGS REFER TO DIRECTIVE (EU) 2022/2555 (NOTABLY ART. 21, 23, AND WHERE APPLICABLE ART. 28). THE MAPPING IS NON-EXHAUSTIVE AND LISTS REPRESENTATIVE CONTROL ANCHORS [19]–[24].

Threat (Table II)	ISO/IEC 27001:2022 Annex A (examples)	SOC 2 TSC (examples)	NIS2 (examples)	Further norms (examples)
P1–T1 Typosquatting / look-alikes	A.5.7; A.6.3; A.8.16; A.8.23; A.8.7; A.5.25–A.5.27	CC3; CC6; CC7; CC9	Art. 21(2)(a)(b)(f)(g)(j); Art. 23	NIST CSF: PR.AT, DE.CM; CIS: 9, 14, 17
P1–T2 Orphaned/published names	A.5.9; A.5.37; A.8.9; A.8.16; A.8.32	CC7; CC8; CC9	Art. 21(2)(a)(e)(f)(i)	NIST CSF: ID.AM; CIS: 1, 2, 4
P2–T1 DNS misconfig	A.8.9; A.8.15–A.8.16; A.8.20–A.8.21; A.8.24; A.8.8	CC7; CC8; CC3	Art. 21(2)(a)(e)(f)(h)	RFC 4033–4035 (DNSSEC); RFC 8659 (CAA); CIS: 4
P2–T2 Subdomain takeover (dangling)	A.5.21–A.5.23; A.8.9; A.8.16; A.8.32	CC7; CC8; CC9; CC6	Art. 21(2)(d)(e)(f)	NIST CSF: GV.SC; CIS: 12, 15, 16
P2–T3 Registrar account compromise	A.5.15–A.5.18; A.8.2; A.8.5; A.8.15–A.8.16; A.5.24–A.5.28	CC6; CC7; CC9	Art. 21(2)(i)(j)(b); Art. 23	NIST SP 800-63-4; CIS: 5, 6, 8, 17
P3–T1 Phishing / fraud on expired domains	A.5.7; A.5.9; A.6.3; A.8.16; A.5.25–A.5.27	CC3; CC7; CC9	Art. 21(2)(a)(b)(f)(g)(j); Art. 23	NIST CSF: DE.*, RS.*; CIS: 9, 14, 17
P3–T2 Email / account takeovers	A.5.16–A.5.18; A.8.5; A.8.15–A.8.16; A.5.25–A.5.27	CC6; CC7	Art. 21(2)(i)(j)(b); Art. 23	NIST SP 800-63-4; CIS: 5, 6, 9, 17
P3–T3 Lingering integrations (OAuth/SSO/API)	A.5.21–A.5.23; A.8.9; A.8.32; A.8.16; A.5.25–A.5.27	CC6; CC7; CC8; CC9	Art. 21(2)(d)(e)(f)(i)(j)	NIST CSF: GV.SC; CIS: 15, 16, 17
P3–T4 Drop-catching / speculation	A.5.7; A.5.9; A.5.31–A.5.32; A.5.37	CC3; CC7; CC9	Art. 21(2)(a)(f); (+ Art. 28 for registration services)	NIST CSF: ID.RA; CIS: 1, 17
P4–T1 Speculation/parking abuse	A.5.31–A.5.32; A.5.7; A.8.16; A.8.23	CC3; CC7; CC9	Art. 21(2)(a)(b)(f); (+ Art. 28 for registration services)	NIST CSF: DE.CM; CIS: 9, 17
P4–T2 Criminal monetization (phishing/C2/spam)	A.8.7; A.8.8; A.8.15–A.8.16; A.8.20–A.8.21; A.5.24–A.5.27	CC6; CC7; CC9	Art. 21(2)(b)(c)(f)(j); Art. 23; Art. 29	NIST CSF: DE.*, RS.*, RC.*; CIS: 8, 10, 13, 17

The following inputs are assumed for each protected object B:

- A set of *canonical domains* $\mathcal{D}_B = \{d_{\text{can}}^{(1)}, \ldots, d_{\text{can}}^{(m)}\}$ representing primary domains, regional domains, and other authoritative domains under the control of the organization.
- A set of *candidate domains* $\mathcal{D}_B^*$ to be evaluated (e.g., newly observed registrations, permutations, TLD variants, or look-alikes).
- Optional, brand-/institution-specific configuration: (i) lists of subbrands, products, or program names; (ii) token lists capturing internal naming conventions and stakeholder roles; (iii) relevant countries, jurisdictions, or language variants; and (iv) optional allowlists for known benign third parties (e.g., long-standing fan communities).
- A sector- and organization-specific multiplier $C_{10}(B) = m_{\text{ind}}(B) \in \mathbb{R}$, encoding the criticality of the protected object B in the organizational risk model (values > 1 amplify, values in $0 < m_{\text{ind}}(B) < 1$ attenuate; values $m_{\text{ind}}(B) \leq 0$ are permitted and are clamped to zero in Eq. (6)).

Category selection methodology and design principles: The criterion set is selected to cover attack-relevant imitation patterns while remaining measurable at scale and interpretable across technical, legal, and organizational stakeholders. Selection proceeds as follows. First, abuse mechanisms relevant to domains are enumerated from the threat landscape (e.g., typosquatting, homoglyph-based confusion, combosquatting, and jurisdictional mimicry). Second, each mechanism is mapped to *observable* properties of $\text{reg}(d)$ that are available at or shortly after registration time, prior to costly content-based analysis. Third, the resulting feature families are filtered by (i) reproducibility (clear definitions, deterministic computation), (ii) robustness (limited sensitivity to minor adversarial perturbations), (iii) cross-domain applicability (private- and public-sector protected identifiers), and (iv) operational feasibility (availability of required configuration data). Finally, criteria that require additional organizational knowledge (e.g., sub-brand portfolios or stakeholder-specific naming conventions) are explicitly marked as *optional*; the scoring function supports systematic exclusion of such criteria without changing the interpretation of the remaining score.

Figure 1 provides an end-to-end overview of the deterministic scoring pipeline, from criteria extraction and normalization to weighted aggregation and the mapping of $\text{Score}(d, B)$ to operational risk bands and response actions.

Weighted aggregation with explicit handling of optional criteria: For each candidate domain $d \in \mathcal{D}_B^*$, the algorithm computes nine domain-level criteria

$$C_1(d, B), \ldots, C_9(d, B) \in [0, 1], \tag{2}$$

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

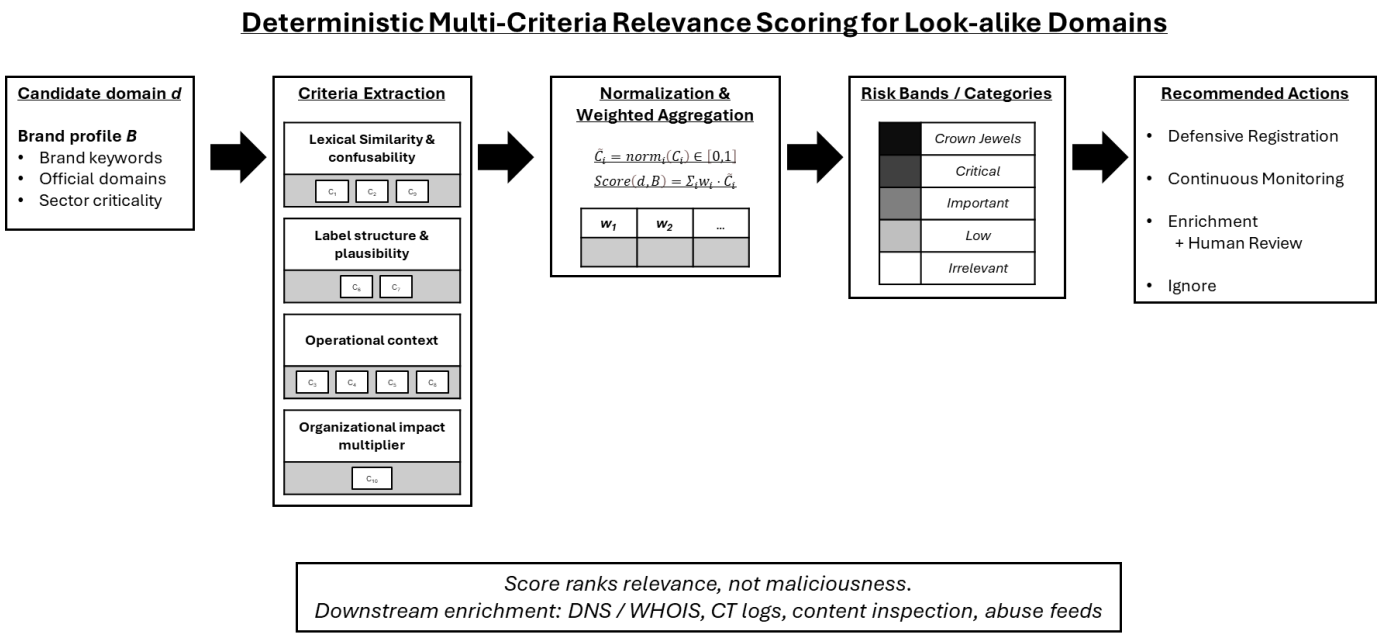


Figure 1. Deterministic multi-criteria relevance scoring pipeline for look-alike domains. Starting from a candidate domain d and a brand profile B, the model extracts criteria C_1–C_{10} grouped into lexical confusability, label structure and plausibility, operational context, and an organizational impact multiplier. Each criterion is normalized to $\tilde{C}_i \in [0, 1]$ and aggregated using fixed weights to obtain $\mathrm{Score}(d, B)$, which is mapped into operational risk bands and corresponding response actions. The score ranks *relevance* rather than maliciousness and is intended to be combined with downstream enrichment (DNS/WHOIS telemetry, certificate transparency logs, content inspection, and abuse feeds). Own depiction.

and aggregates them into a base relevance value using a weighted mean. Let $I(B) \subseteq \{1, \ldots, 9\}$ denote the set of *active* criteria with available required inputs for B. The base relevance is defined as

$$R_{\mathrm{base}}(d, B) = \frac{\sum_{i \in I(B)} w_i \cdot C_i(d, B)}{\sum_{i \in I(B)} w_i}, \tag{3}$$

where $w_i > 0$ are criterion weights. If a criterion is disabled or unavailable (e.g., C_3 and C_4 when no keyword or subbrand lists exist), it is excluded from both numerator and denominator via $I(B)$, which preserves comparability and avoids implicit penalization due to missing organizational metadata.

In the absence of labeled ground truth for a specific organization, a conservative default is to emphasize confusability-driven criteria and to down-weight optional semantic criteria. Table IV provides one reproducible baseline configuration; weights can be calibrated using (i) labeled domain abuse corpora, (ii) historical incident data, or (iii) structured expert elicitation (e.g., pairwise comparisons or Delphi-style consensus). Regardless of calibration method, transparent reporting of w_i is necessary to enable replication and interpretation.

Brand-relative normalisation: Absolute base scores are not directly comparable across different protected identifiers, because identifier length, linguistic structure, and portfolio breadth affect baseline similarity. To support cross-brand ranking and consistent interpretation, scores are normalized relative to the canonical domains of B. Define the *brand anchor*

$$A(B) = \max_{d' \in \mathcal{D}_B} R_{\mathrm{base}}(d', B), \tag{4}$$

TABLE IV
ILLUSTRATIVE DEFAULT WEIGHTS AND OPTIONALITY. CRITERIA C_1–C_2, C_5–C_7 ARE COMPUTABLE FROM THE DOMAIN STRING ALONE AND ARE TREATED AS CORE. CRITERIA C_3, C_4, AND C_8 REQUIRE BRAND-/INSTITUTION-SPECIFIC CONFIGURATION AND ARE OPTIONAL. OWN ILLUSTRATIVE CONFIGURATION.

Criterion	Default weight w_i
C_1 Lexical similarity	3
C_2 Visual/IDN similarity	2
C_3 Functional/stakeholder semantics*	1
C_4 Subbrand/product coverage*	1
C_5 TLD risk and typo-proximity	2
C_6 Lexical complexity prior	1
C_7 Token structure and order	2
C_8 Geo-/market semantics*	1
C_9 Brand genericity adjustment	1

* Optional Criteria.

i.e., the maximum base relevance assigned to authoritative domains. The brand-relative score is then

$$R_{\mathrm{rel}}(d, B) = \begin{cases} 0, & \text{if } A(B) = 0, \\ \min\left(1, \dfrac{R_{\mathrm{base}}(d, B)}{A(B)}\right), & \text{otherwise.} \end{cases} \tag{5}$$

By construction, $R_{\mathrm{rel}}(d, B) \approx 1$ for canonical domains and near-canonical variants, while smaller values indicate proportionally less relevance.

Organizational multiplier and 0–100 scale: The domain-level criteria represent a technical and semantic approximation of confusability and association. Organizational risk management can require different operating points across sectors and

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

assets. This is captured by a multiplier $C_{10}(B) = m_{\text{ind}}(B) \in \mathbb{R}$, for example to prioritize domains related to critical infrastructure operators, high-value digital identities, or essential public services. The multiplier is applied after brand-relative normalization:

$$R_{\text{final}}(d, B) = \min\big(1,\ \max(0,\ R_{\text{rel}}(d, B) \cdot m_{\text{ind}}(B))\big). \quad (6)$$

Finally, the score is mapped to the operational range $[0, 100]$ as

$$\text{Score}(d, B) = \lfloor 100 \cdot R_{\text{final}}(d, B) + 0.5 \rfloor, \quad (7)$$

i.e., integer rounding; continuous scores can be retained if required for fine-grained ranking.

Pseudocode: Figure 2 summarizes the computation. The anchor $A(B)$ depends only on B and the chosen criterion configuration; it can be precomputed once per protected object and cached. It requires the Brand B, the set of related owned domains D_B, the candidate domain d, the weights $w_1, ..., w_9$ and an industry multiplier $m_ind(B)$ in R as input. The output will be a score between 0 and 100.

V. Category Definitions

This section defines criteria C_1–C_{10}. Criteria C_1–C_9 are domain-level values in $[0, 1]$ and are aggregated as in Eq. (3). Criterion $C_{10}(B)$ is a brand-/institution-level multiplier applied in Eq. (6). Unless explicitly stated, criteria operate on the registrable label $\ell(d)$ (i.e., without the TLD); C_5 is the only criterion that explicitly incorporates $\tau(d)$ or cross-TLD typo-proximity.

A. C1: Lexical brand similarity (string similarity)

Optionality: core (not optional). **Scope:** label $\ell(d)$ only.

Criterion $C_1(d, B)$ captures lexical similarity between the candidate label and the protected identifier strings in $\mathcal{B}_B$. Let $\text{norm}(\cdot)$ denote deterministic normalization (case-folding, Unicode Normalization Form KC (NFKC), and removal of separators such as hyphens and dots). Let $\text{DL}(\cdot, \cdot)$ denote Damerau–Levenshtein distance (insertions, deletions, substitutions, and transpositions) [49], [50]. A normalized similarity is

$$\text{sim}_{\text{DL}}(x, y) = 1 - \frac{\text{DL}(x, y)}{\max\{|x|, |y|\}}, \quad (8)$$

and the criterion is defined as

$$C_1(d, B) = \max_{b \in \mathcal{B}_B} \text{sim}_{\text{DL}}\big(\text{norm}(b), \text{norm}(\ell(d))\big). \quad (9)$$

The formulation is intentionally modular: other lexical similarity functions can be substituted without changing the overall algorithmic structure, including Levenshtein distance, Jaro–Winkler, cosine similarity over character n-grams, or token-based similarity (e.g., Jaccard similarity) [51], [52] when $\ell(d)$ contains multiple tokens. The choice affects sensitivity to transpositions, prefixes/suffixes, and short strings, and should be reported explicitly.

Example. For B = "ExampleBank" and d = `examp1ebank-login.example`, the label $\ell(d)$ = `examp1ebank-login` differs by a single substitution (`1` → `l`) and an added token (`login`), leading to $C_1(d, B)$ close to 1.

```
Brand-relative domain relevance score

# parse registrable domain and tld
(ell, tau) <- parse_registrable(d)

# determine active criteria
I <- {i in {1,...,9} | crit_i_available(B)}

# compute criteria C_1,...,C_9
for i in I do
  C_i(d,B) <- compute_criterion_i(ell, tau, B)
end for

# weighted aggregation over active criteria
R_base(d,B) <- (
      sum_{i in I} w_i * C_i(d,B)
  ) / (sum_{i in I} w_i)

# compute brand anchor A(B) if not cached
if anchor_not_cached(B) then
  A(B) <- 0
  for each d_can in D_B do
    (ell_can, tau_can) <- parse_reg(d_can)
    for i in I do
      C_i(d_can,B) <- compute_crit_i(
        ell_can, tau_can, B)
    end for
    R_base_can <- (
        sum_{i in I} w_i * C_i(d_can,B)
    ) / (sum_{i in I} w_i)
    if R_base_can > A(B) then
      A(B) <- R_base_can
    end if
  end for
  cache_anchor(B, A(B))
else
  A(B) <- cached_anchor(B)
end if

# brand-relative normalisation
if A(B) == 0 then
    R_rel(d,B) <- 0
else
    R_rel(d,B) <- min(1, R_base(d,B) / A(B))
end if

# apply org-multiplier and map to [0,100]
R_final(d,B) <- min(1, max(0, R_rel(d,B)
  * m_ind(B)))
Score(d,B)   <- round(100 * R_final(d,B))
return Score(d,B)
```

Figure 2. Deterministic computation of a brand-relative relevance score for a candidate domain d with respect to protected object B. Own depiction.

B. C2: Visual / IDN confusion and homoglyphs

Optionality: core (not optional). **Scope:** label $\ell(d)$ only.

Lexical similarity alone does not capture visually confusable Internationalized Domain Names (IDNs) and homoglyph substitutions (e.g., Cyrillic characters resembling Latin characters). Criterion $C_2(d, B)$ models visual confusability by comparing *visual skeletons*. Let $\text{skel}(\cdot)$ map a Unicode string to a

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

canonical skeleton via a confusable-character mapping (e.g., as standardized in Unicode security guidance). The criterion uses the same normalized similarity as C_1 on skeletons:

$$C_2(d, B) = \max_{b \in \mathcal{B}_B} \mathrm{sim}_{\mathrm{DL}}\big(\mathrm{skel}(\mathrm{norm}(b)), \mathrm{skel}(\mathrm{norm}(\ell(d)))\big). \tag{10}$$

This criterion is particularly important in multilingual and public-sector settings, where official designations are commonly targeted via cross-script substitutions.

Example. For $B =$ "ExampleBank" and a candidate label that replaces Latin "a" with visually similar Cyrillic "а", the string may be lexically distant but visually indistinguishable; $C_2(d, B)$ captures this through identical skeletons.

C. C3: Functional and stakeholder semantics

Optionality: optional. **Scope:** label $\ell(d)$ only.

Criterion $C_3(d, B)$ measures whether $\ell(d)$ contains tokens that suggest functions, services, or stakeholder groups typically associated with B. This criterion captures the observation that many high-impact look-alikes include terms such as `login`, `verify`, `support`, `billing`, `mail`, or (public-sector) `portal`, `services`, `tax`, `police`. Let $\mathrm{tok}(\ell(d))$ split the label into tokens (e.g., along hyphens and alphanumeric boundaries). Let K^{func} and K^{stake} denote configurable keyword sets with weights $\alpha_k \in (0, 1]$ reflecting their indicative strength. Define

$$s_{\mathrm{func}}(d) = \max_{k \in K^{\mathrm{func}}} \alpha_k \cdot \mathbf{1}[k \in \mathrm{tok}(\ell(d))], \tag{11}$$

$$s_{\mathrm{stake}}(d) = \max_{k \in K^{\mathrm{stake}}} \alpha_k \cdot \mathbf{1}[k \in \mathrm{tok}(\ell(d))], \tag{12}$$

and set

$$C_3(d, B) = \min\big(1,\ s_{\mathrm{func}}(d) + s_{\mathrm{stake}}(d)\big). \tag{13}$$

If no keyword sets are available or desired, C_3 is excluded via $I(B)$ in Eq. (3).

Example. For $B =$ "ExampleBank" and $d =$ `examplebank-customer-login.example`, the tokens {`customer`, `login`} match both stakeholder and function lists, yielding a high $C_3(d, B)$.

D. C4: Brand and product coverage (subbrands, services)

Optionality: optional. **Scope:** label $\ell(d)$ only.

Many organizations operate multiple product lines, subbrands, or program names that are independently targeted in domain abuse. Criterion $C_4(d, B)$ captures whether $\ell(d)$ contains references to such assets. Let $\mathcal{S}_B$ be a configurable set of subbrand/product identifiers (strings or tokens). Define a token-level similarity between a subbrand s and a label token set as

$$\mathrm{sim}_{\mathrm{tok}}(s, \ell(d)) = \max_{t \in \mathrm{tok}(\ell(d))} \mathrm{sim}_{\mathrm{DL}}(\mathrm{norm}(s), \mathrm{norm}(t)), \tag{14}$$

and set

$$C_4(d, B) = \max_{s \in \mathcal{S}_B} \mathrm{sim}_{\mathrm{tok}}(s, \ell(d)). \tag{15}$$

If $\mathcal{S}_B$ is unavailable, C_4 is excluded via $I(B)$.

Example. For $B =$ "Fabrikam" with subbrand list $\mathcal{S}_B =$ {`CloudDrive`, `Pay`}, the domain `fabrikam-clouddrive.example` yields $C_4(d, B)$ close to 1 due to the explicit subbrand reference.

E. C5: TLD risk, market relevance, and typo-proximity

Optionality: core (not optional). **Scope:** includes TLD $\tau(d)$ and registrable domain $\mathrm{reg}(d)$.

Criterion $C_5(d, B)$ incorporates two complementary aspects that cannot be captured from $\ell(d)$ alone: (i) the relevance and risk associated with the TLD choice, and (ii) the observation that near-duplicate domains of canonical registrations (including cross-TLD variants) are frequently relevant due to user typos and adversarial imitation.

TLD relevance and baseline risk. Let $r_{\mathrm{mkt}}(\tau, B) \in [0, 1]$ encode the market or jurisdiction relevance of a TLD τ for B (e.g., ccTLDs of operating regions, sector-specific TLDs), and let $r_{\mathrm{abuse}}(\tau) \in [0, 1]$ encode a configurable baseline abuse propensity of τ (obtainable from measurement or cyber threat intelligence (CTI) feeds). A simple combination is

$$r_{\mathrm{tld}}(d, B) = \max\big(r_{\mathrm{mkt}}(\tau(d), B),\ r_{\mathrm{abuse}}(\tau(d))\big). \tag{16}$$

This construction avoids over-penalizing a TLD that is operationally relevant for B even if baseline abuse is low, and conversely flags TLDs with elevated abuse even when market relevance is unknown.

Typosquatting and near-duplicate proximity. Let $\mathrm{DL}_{\mathrm{reg}}(\cdot, \cdot)$ be an edit distance over $\mathrm{reg}(d)$ (including both ℓ and τ), optionally with keyboard-adjacency weights and common TLD error patterns (e.g., `.com` ↔ `.co`). Define a similarity-derived proximity score

$$p_{\mathrm{typo}}(d, B) = \max_{d_{\mathrm{can}} \in \mathcal{D}_B} \exp\big(-\lambda \cdot \mathrm{DL}_{\mathrm{reg}}(\mathrm{reg}(d), \mathrm{reg}(d_{\mathrm{can}}))\big), \tag{17}$$

with $\lambda > 0$ controlling the decay. This term assigns high values to domains that are plausible typos or close variants of authoritative domains, thereby ensuring that domains with a high probability of user confusion (or adversarial imitation) receive an elevated relevance score even when the lexical similarity of $\ell(d)$ alone is ambiguous.

The combined criterion is then

$$C_5(d, B) = \max\big(r_{\mathrm{tld}}(d, B),\ p_{\mathrm{typo}}(d, B)\big). \tag{18}$$

Example. If $\mathcal{D}_B$ contains `examplebank.com`, then `examplebank.co` (a one-character TLD omission) yields a high $p_{\mathrm{typo}}(d, B)$ and therefore a high $C_5(d, B)$, even if `.co` has unknown market relevance for B.

F. C6: Lexical complexity prior

Optionality: core (not optional). **Scope:** label $\ell(d)$ only.

Criterion $C_6(d, B)$ provides a weak prior against unrelated random-looking labels that are unlikely to be operationally relevant, while remaining non-dominant when strong brand similarity exists. Let $H(\ell)$ denote the Shannon entropy [53] of characters in $\mathrm{norm}(\ell)$, and normalize it as

$$H_{\mathrm{norm}}(\ell) = \frac{H(\ell)}{\log_2(|\Sigma|)}, \tag{19}$$

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

where Σ is the character alphabet observed in $\mathrm{norm}(\ell)$. The logarithmic normalisation reflects the fact that Shannon entropy is bounded by $\log_2(|\Sigma|)$, ensuring that entropy values are comparable across different alphabets and string representations. Let

$$q_{\mathrm{dig}}(\ell) = \frac{\#\{\text{digits in } \mathrm{norm}(\ell)\}}{|\mathrm{norm}(\ell)|} \tag{20}$$

denote the digit ratio. A simple reproducible definition is

$$C_6(d, B) = 1 - \min\left(1,\ \frac{H_{\mathrm{norm}}(\ell(d)) + q_{\mathrm{dig}}(\ell(d))}{2}\right). \tag{21}$$

Low-entropy, human-readable labels yield values near 1; highly random strings yield values near 0. In the aggregation, C_6 should be assigned a comparatively small weight to avoid suppressing high-similarity typosquats.

Example. `examplebank-help.example` has low entropy and no digit noise, leading to high C_6; `x9k3q7p2.example` yields low C_6.

G. C7: Structure and position of the brand token

Optionality: core (not optional). **Scope:** label $\ell(d)$ only.

Beyond character-level similarity, attackers frequently exploit token structure: inserting functional tokens (`login`, `secure`), moving the brand token, or swapping token order. Criterion $C_7(d, B)$ compares the token sequence of $\ell(d)$ to token sequences observed in canonical domains. Let $T(d) = \mathrm{tok}(\ell(d))$ be the token sequence of the candidate label, and let $T(d_{\mathrm{can}})$ be the token sequence of a canonical domain. Define a token-level Damerau–Levenshtein distance $\mathrm{DL}_{\mathrm{tok}}(\cdot,\cdot)$ over token sequences (allowing token transpositions). A normalized similarity is

$$\mathrm{sim}_{\mathrm{tok}}(T, T') = 1 - \frac{\mathrm{DL}_{\mathrm{tok}}(T, T')}{\max\{|T|, |T'|\}}, \tag{22}$$

and the criterion is

$$C_7(d, B) = \max_{d_{\mathrm{can}} \in \mathcal{D}_B} \mathrm{sim}_{\mathrm{tok}}\big(T(d), T(d_{\mathrm{can}})\big). \tag{23}$$

This construction preserves high relevance under token swaps (single transpositions) while still distinguishing heavily restructured domains.

Example (token swap). If $\mathcal{D}_B$ contains `northwind-payments.example` with token sequence [`northwind`, `payments`], then `payments-northwind.example` differs by one token transposition and therefore retains a high $C_7(d, B)$, albeit lower than an exact match.

H. C8: Geo- and market semantics in the label

Optionality: optional. **Scope:** label $\ell(d)$ only.

Many high-relevance domains embed geographic or jurisdictional markers (e.g., country codes, city names, language identifiers). Criterion $C_8(d, B)$ captures whether such markers align with the operating footprint of B. Let $\mathcal{G}_B$ be a configurable set of geo tokens relevant to B (e.g., `uk`, `de`, `berlin`). Let $\mathcal{G}$ be a broader geo vocabulary. A deterministic rule-based scoring is

$$C_8(d, B) = \begin{cases} 1, & \text{if } \mathrm{tok}(\ell(d)) \cap \mathcal{G}_B \neq \emptyset, \\ 0, & \text{if } \mathrm{tok}(\ell(d)) \cap (\mathcal{G} \setminus \mathcal{G}_B) \neq \emptyset, \\ 0.5, & \text{otherwise.} \end{cases} \tag{24}$$

If no geo configuration exists, C_8 is excluded via $I(B)$.

Example. For an organization with footprint $\mathcal{G}_B = \{\mathtt{uk}, \mathtt{ie}\}$, the domain `examplebank-uk.example` yields $C_8 = 1$, whereas `examplebank-br.example` yields $C_8 = 0$ if `br` is outside the footprint.

I. C9: Brand genericity and dictionary overlap

Optionality: core (not optional). **Scope:** protected identifier B (brand-level adjustment used at domain level).

Some protected identifiers are dictionary words or highly generic strings, which increases the rate of benign coincidental matches and can inflate lexical similarity scores. Criterion $C_9(d, B)$ adjusts for this by down-weighting generic identifiers. Let $g(B) \in [0, 1]$ denote a genericity score, for example derived from (i) a dictionary membership test, (ii) corpus frequency of the canonical identifier token, or (iii) a learned language-model prior. Higher values indicate more generic terms. The criterion is defined as

$$C_9(d, B) = 1 - g(B). \tag{25}$$

As a result, coined or rare identifiers (small $g(B)$) preserve high overall relevance for similar domains, whereas generic identifiers yield more conservative relevance scores.

Example. If B is a coined identifier (e.g., `Xyphora`) with $g(B) \approx 0$, then $C_9 \approx 1$. If B is a dictionary word with high frequency, $g(B)$ is larger and C_9 reduces the aggregate relevance of weakly matching domains.

J. C10: Organizational criticality and sector (multiplier)

Optionality: optional (policy-driven). **Scope:** protected object B only.

Criterion $C_{10}(B) = m_{\mathrm{ind}}(B) \in \mathbb{R}$ is a multiplier that expresses organizational criticality and sector-specific risk posture. Unlike C_1–C_9, it is not derived from the domain string; it represents a policy input tied to the protected object and the operational impact of abuse. Typical assignments set $m_{\mathrm{ind}}(B) = 1$ as default, use moderate amplification (e.g., 1.1 to 1.3) for high-impact contexts, and allow attenuation (e.g., 0.7 to 0.9) for lower-criticality protected objects; values $m_{\mathrm{ind}}(B) \leq 0$ are permitted but collapse to zero after clamping in Eq. (6). The final score is capped through Eq. (6) to preserve the $[0, 1]$ normalization prior to the 0–100 mapping.

Example. Two candidate domains with identical $R_{\mathrm{rel}}(d, B)$ can receive different final scores if one protected object represents a high-criticality identity portal (e.g., a public eID service) and is assigned a higher $m_{\mathrm{ind}}(B)$ than an auxiliary informational subbrand.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

K. False Positives, False Negatives, and Organizational Trade-offs

The scoring function is designed to rank *relevance* rather than *maliciousness*; therefore, misclassifications are inevitable and must be handled operationally. In particular, three recurring sources of trade-offs should be made explicit:

- **Generic protected terms (e.g., "bank").** If the protected identifier contains dictionary words or generic tokens, lexical similarity can generate substantial benign background noise. Criterion C_9 mitigates this effect through genericity adjustment, but organizations should still expect more false positives and may need higher thresholds, stricter allowlists, or additional context features for enforcement decisions.
- **Fan and criticism sites.** Domains that intentionally refer to B (press portals, communities, criticism/watchdog pages) may legitimately obtain high relevance scores. For security monitoring, this is useful because such domains can still be abused (e.g., via compromise or ads/redirects). For takedown or legal enforcement, however, the score is insufficient without content review and contextual assessment.
- **Legal vs. technical assessment.** The score provides a deterministic technical triage signal and is not a substitute for trademark adjudication or jurisdiction-specific legal analysis. Actions such as defensive registration, UDRP proceedings, registrar escalation, or takedown requests require additional evidence (e.g., website content, intent signals, disclaimers, and jurisdiction), and should be routed through a defined workflow between security, legal, and communications stakeholders.

Operationally, organizations should use the score as an upstream prioritization mechanism and combine it with downstream enrichment (WHOIS/DNS telemetry, certificate transparency signals, content inspection, and abuse feeds) and a human-in-the-loop review for borderline cases. Thresholds should be calibrated to analyst capacity and risk appetite, often separating a lower threshold for monitoring from a higher threshold for enforcement.

VI. Discussion, Limitations, and Research Agenda

This extended version combines a life-cycle-based threat model (Section III) with a deterministic, multi-criteria relevance scoring function for look-alike domains (Sections IV–V). A central implication is that domain security is often treated as a *static* configuration and ownership problem, even though the most damaging incidents emerge from *transitions* across the domain life cycle (e.g., loss of control, re-registration, and residual trust). Table V contrasts these perspectives and highlights why a phase-based program requires different governance and metrics than a snapshot-based posture assessment.

A. Limitations

Three limitations are particularly relevant for interpreting the contributions of this work and for designing a validation roadmap.

No ground truth. Both the life-cycle threat mapping and the relevance-scoring model are derived from literature, incidents, and threat intelligence rather than from a labeled, organization-specific ground truth dataset. Consequently, the default weights in Table IV and the risk-band thresholds in Section IV are illustrative rather than empirically calibrated. This limits quantitative claims about predictive performance and requires careful interpretation: the score is best understood as a reproducible prioritization heuristic until validated against abuse labels and operational outcomes.

No content inspection. The proposed model deliberately avoids content-based analysis to remain applicable early in the domain life cycle (e.g., at registration time) and to scale across large candidate spaces. As a result, the score cannot determine malicious intent or classify the nature of abuse. High-scoring domains can be benign (e.g., press portals, authorized resellers, fan/criticism sites), and some malicious domains may avoid strong lexical similarity. Practical use therefore requires downstream enrichment (WHOIS/DNS telemetry, certificate transparency signals, web content analysis, and abuse feeds) and clear workflows for escalation and enforcement.

No subdomain semantics. The scoring function operates on the registrable domain $\mathrm{reg}(d)$ to support registration and monitoring decisions at the domain-registration level. However, many real-world attacks exploit *subdomain*-level semantics and delegation (e.g., subdomain takeover, dangling CNAMEs, and service-specific hostnames). Similarly, phishing often occurs on attacker-controlled subdomains and URL paths even when the registrable domain appears benign. Capturing these aspects requires additional DNS- and service-level semantics beyond the domain string and is out of scope for the current deterministic scoring definition.

B. Research Directions

The following research directions translate the above limitations into a concrete agenda for empirical validation and operational adoption.

a) Empirical correlation between relevance score and observed abuse: A first validation step is to quantify the relationship between $\mathrm{Score}(d, B)$ and downstream abuse outcomes. This requires constructing datasets that link (i) candidate domains (e.g., newly registered domains, permutations, and observed look-alikes) with (ii) abuse labels from phishing, malware, spam, and fraud reporting sources, as well as incident response records where available. Analyses should evaluate ranking quality (e.g., precision at k, PR/ROC curves), calibration (score bands vs. abuse probability), and time-to-abuse dynamics (how quickly high-scoring domains transition into abuse). Results can inform both threshold selection and weight calibration for different sectors and organizational risk appetites.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE V
COMPARISON OF LIFECYCLE-BASED VS. STATIC DOMAIN SECURITY MODELS. OWN SYNTHESIS.

Aspect	Static domain security model (snapshot view)	Lifecycle-based model (this work)
Primary perspective	Domains are treated as assets under current ownership; security is mainly evaluated via present-time configuration and access controls.	Domains are treated as assets that transition through phases (pre-registration, active operation, expiry, re-registration); threats and controls are phase-specific.
Threat coverage	Emphasis on active-domain misconfigurations and account compromise; limited treatment of expiry and post-expiry abuse.	Explicitly includes pre-registration look-alikes, operational misconfigurations, expiry governance failures, and post-expiry adversarial monetization.
Control focus	One-off hardening and periodic audits (e.g., DNSSEC, registrar MFA, occasional zone checks).	Continuous governance across phases: monitoring/claiming look-alikes, zone hygiene, renewal and retirement checklists, and detection/response for re-registered identifiers.
Typical blind spots	Residual trust after expiry; lingering OAuth/SSO callbacks; attacker re-registration; inherited reputation effects.	Still requires downstream enrichment to distinguish benign from malicious uses; depends on operational integration and empirical calibration.
Operational metrics	Compliance posture metrics (e.g., DNSSEC coverage, MFA enabled) and point-in-time audit results.	Phase-specific KPIs: coverage of defensive registrations, renewal success rate, dangling record resolution, detection latency for re-registrations, and response time.

b) Cost-benefit models for defensive registration and monitoring: A second direction is to formalize the economic and operational trade-offs of defensive registration strategies. The score provides a structured way to define candidate sets and priority bands, but organizations ultimately need decision rules that balance recurring registration costs, monitoring/analyst effort, and expected loss from abuse (including reputational impact). Models can incorporate (i) expected abuse probability per score band, (ii) expected incident cost distributions, and (iii) operational capacity constraints, yielding quantitative guidance such as "register all domains above threshold t", "monitor band t_1 to t_2", or "ignore below t_2". Such models are especially important for SMEs that cannot sustain broad defensive portfolios.

c) Integration into CTI pipelines and operational tooling: A third direction is the integration of the life-cycle model and the scoring function into CTI workflows. Concretely, organizations can (i) ingest newly observed registrations and candidate permutations, (ii) compute $\mathrm{Score}(d, B)$ as an upstream filter, (iii) enrich prioritized domains with DNS/WHOIS/certificate telemetry and content signals, and (iv) connect results to playbooks (defensive registration, blocking, takedown escalation, or monitoring). Validation should study operational outcomes such as analyst workload reduction, time-to-detection, and reduction of successful impersonation events.

C. Practical Adoption Considerations

While the proposed model is intentionally deterministic, practical deployment depends on organizational context and stakeholder roles.

a) SMEs vs. enterprises: Enterprises typically operate larger domain portfolios, have dedicated security/brand teams, and can maintain richer configuration inputs (keyword lists, subbrands, geo footprints). SMEs often face stricter resource constraints and benefit from a simplified configuration (core criteria only) combined with managed monitoring services. For SMEs, conservative thresholds and a smaller, high-confidence candidate set can be preferable to broad monitoring that creates alert fatigue.

b) Registrars vs. brand owners: Brand owners can deploy the model for pre-registration monitoring and defensive acquisition decisions, while registrars and registries can leverage lifecycle awareness to reduce systemic risk (e.g., secure defaults, clearer expiry/renewal governance, and detection of misconfigurations that enable takeover scenarios). This division of responsibilities matters because some high-impact controls (e.g., account recovery hardening, registry lock offerings, or misconfiguration handling) are primarily registrar-side, whereas look-alike monitoring and takedown escalation are often brand-side.

c) Public sector particularities: Public-sector identifiers (ministries, agencies, citizen portals, and digital identity services) exhibit high impersonation risk and multilingual naming patterns. Here, relevance scoring must account for abbreviations, language variants, and cross-script confusion, and adoption should be coupled with citizen-facing guidance (e.g., official domain lists) and coordination with national Computer Emergency Response Team / Computer Security Incident Response Team (CERT/CSIRT) processes. Additionally, public-sector enforcement pathways may differ from private trademark-driven processes, making clear triage boundaries (monitoring vs. legal action) particularly important.

VII. CONCLUSION AND FUTURE WORK

Domains are critical security assets; a life cycle view shows distinct weaknesses before registration, during operation, after expiration, and after re-registration.

Derivation: The conclusions synthesize the phase-wise mapping in Table II with the structured literature scan in Section II, thereby answering **RQ1** and **RQ2**. Sections IV–V address **RQ3** by formalizing a deterministic relevance score for look-alike domains, and Section VI translates remaining gaps into a research agenda for empirical validation (**RQ4**).

Implications: Domain security must be run as a continuous program across DNS configuration and registrar governance; safer defaults by registrars, registries, and CAs reduce systemic risk.

In addition, the deterministic relevance-scoring model for domain names introduced in Sections IV–V can help organi-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

zations classify look-alike domains into those to defensively register, those to monitor, and those to safely ignore, thereby making passive domain registration strategies more transparent and measurable.

a) Recommendations: Based on the phase-wise mapping in Table II, we recommend prioritizing DNSSEC deployment for apex and critical zones, constraining certificate issuance via CAA records, and hardening registrar accounts (multi-factor authentication (MFA), registry lock, and two-party recovery). Operationally, these controls should be complemented by scheduled zone hygiene (e.g., dangling-record resolution and takeover scans) as well as renewal and deprovisioning governance to reduce expiry-driven loss-of-control and residual-trust exposure.

b) Limitations: This study is a conceptual, literature-based synthesis and does not contribute new measurement data. The evidence base is dominated by English-language sources from 2014–2025, and prevalence estimation is out of scope. Accordingly, the proposed relevance score should be interpreted as a transparent prioritization heuristic rather than a validated predictor of abuse.

c) Future work: A key next step is to empirically test, validate, and (where necessary) calibrate the proposed relevance-scoring algorithm (Sections IV–V) against ground-truth abuse labels and operational outcomes; this evaluation is explicitly *not* part of the present work. In addition, future studies should (i) measure DNSSEC/CAA adoption and misconfiguration at scale, (ii) analyze WHOIS- and DNS-based re-registration and abuse patterns, (iii) assess registrar-security defaults and their impact on takeover risk, and (iv) audit SaaS/OAuth residuals during domain retirement to quantify residual-trust exposure.

Overall, treating domains as lifecycle-managed assets and complementing this view with transparent relevance scoring provides a practical foundation for measurable domain governance, while rigorous empirical validation remains an open task.

REFERENCES

[1] T. Fritzler and M. Massoth, "The domain name life cycle as an attack surface: Systematic threat mapping and defense recommendations," in *Proceedings of CYBER 2025, The Tenth International Conference on Cyber-Technologies and Cyber-Systems*. Lisbon, Portugal: IARIA Press, 9 2025, pp. 1–7, accessed: 2026.05.25. [Online]. Available: https://www.thinkmind.org/library/CYBER/CYBER_2025/cyber_2025_1_10_80054.html

[2] B. V. Adjibi, A. Avgeditis, M. Antonakakis, M. Bailey, and F. Monrose, "The guardians of name street: Studying the defensive registration practices of the fortune 500," in *Proceedings of the 32nd Network and Distributed System Security Symposium (NDSS 25)*. San Diego, CA, USA: Internet Society, 2 2025.

[3] Y. Zhang, M. Zhang, B. Liu, Z. Liu, J. Zhang, H. Duan, M. Zhang, F. Shi, and C. Xu, "Cross the zone: Toward a covert domain hijacking via shared DNS infrastructure," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, 8 2024, pp. 5751–5768, accessed: 2026.05.25. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/zhang-yunyi-zone

[4] S. Neupane, G. Holmes, E. Wyss, D. Davidson, and L. De Carli, "Beyond typosquatting: an in-depth look at package confusion," in *Proceedings of the 32nd USENIX Conference on Security Symposium*, ser. SEC '23. USA: USENIX Association, 2023.

[5] J. Szurdi, B. Kocsó, G. Cseh, J. M. Spring, M. Félegyházi, and C. Kanich, "The long "taile" of typosquatting domain names," in *Proceedings of the 23rd USENIX Security Symposium (USENIX Security '14)*. San Diego, CA, USA: USENIX Association, Aug. 2014, pp. 191–206, accessed: 2026.05.25. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-szurdi.pdf

[6] J. Friess, T. Gattermayer, N. Gelernter, H. Schulmann, and M. Waidner, "Cloudy with a Chance of Cyberattacks: Dangling Resources Abuse on Cloud Platforms," in *Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI '24)*. USENIX Association, 2024, pp. 1977–1994, accessed: 2026.05.25. [Online]. Available: https://www.usenix.org/conference/nsdi24/presentation/friess

[7] D. Liu, S. Hao, and H. Wang, "All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. New York, NY, USA: Association for Computing Machinery, 10 2016, pp. 1414–1425, accessed: 2026.05.25. [Online]. Available: https://doi.org/10.1145/2976749.2978387

[8] M. Squarcina, M. Tempesta, L. Veronese, S. Calzavara, and M. Maffei, "Can I take your subdomain? exploring Same-Site attacks in the modern web," in *30th USENIX Security Symposium (USENIX Security 21)*. Vancouver, B.C., Canada: USENIX Association, 8 2021, pp. 2917–2934, accessed: 2026.05.25. [Online]. Available: https://www.usenix.org/system/files/sec21-squarcina.pdf

[9] G. Schmid, "Thirty years of DNS insecurity: Current issues and perspectives," *IEEE Communications Surveys & Tutorials*, pp. 2429–2459, 8 2021.

[10] A. Ramdas and R. Muthukrishnan, "A survey on DNS security issues and mitigation techniques," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 781–784.

[11] A. Affinito, R. Sommese, G. Akiwate, S. Savage, K. Claffy, G. M. Voelker, A. Botta, and M. Jonker, "Domain name lifetimes: Baseline and threats," in *Proceedings of the 6th edition of the Network Traffic Measurement and Analysis Conference (TMA Conference 2022)*. International Federation for Information Processing (IFIP), 6 2022, accessed: 2026.05.25. [Online]. Available: https://www.sysnet.ucsd.edu/~voelker/pubs/domain-lifetimes-tma22.pdf

[12] G. C. M. Moura, T. Daniels, M. Bosteels, S. Castro, M. Müller, T. Wabeke, T. van den Hout, M. Korczyński, and G. Smaragdakis, "Characterizing and mitigating phishing attacks at ccTLD scale," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 2147–2161.

[13] M. Wang, X. Zang, J. Cao, B. Zhang, and S. Li, "PhishHunter: Detecting camouflaged IDN-based phishing attacks via siamese neural network," *Computers & Security*, p. 103668, 3 2024, accessed: 2026.05.25. [Online]. Available: https://doi.org/10.1016/j.cose.2023.103668

[14] A. Maurya and S. Sharma, "The art of domain deception: Bifrost's new tactic to deceive users," https://unit42.paloaltonetworks.com/new-linux-variant-bifrost-malware/, 2 2024, accessed: 2026.05.25.

[15] Insikt Group, "SOLARDEFLECTION: C2 Infrastructure Used by NOBELIUM in Company Brand Misuse," Recorded Future, Tech. Rep., 5 2022, accessed: 2026.05.25. [Online]. Available: https://web.archive.org/web/20220503133835/https://go.recordedfuture.com/hubfs/reports/cta-2022-0503.pdf

[16] S. Tilborghs and G. Ferreira, "Flagging 13 million malicious domains in 1 month with newly observed domains," *Akamai Security Blog*, 9 2022, accessed: 2026.05.25. [Online]. Available: https://www.akamai.com/blog/security-research/newly-observed-domains-discovered-13-million-malicious-domains

[17] B. Krebs. (2025, 1) MasterCard DNS error went unnoticed for years. Accessed: 2026.05.25. [Online]. Available: https://krebsonsecurity.com/2025/01/mastercard-dns-error-went-unnoticed-for-years/

[18] L. Nguyen. (2024, 7) BYD-domain führt in deutschland zu sexspielzeug [BYD domain leads to sex toys in germany]. Accessed: 2026.05.25. [Online]. Available: https://sz.de/lux.J5FGj79Eq9c5ThRgCzKaqM

[19] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), "ISO/IEC 27001:2022 information security, cybersecurity and privacy protection — information security management systems — requirements," International Standard, 2022, edition 3 (October 2022); Accessed: 2026.05.25. [Online]. Available: https://www.iso.org/standard/27001

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[20] ——, "ISO/IEC 27002:2022 information security, cybersecurity and privacy protection — information security controls," International Standard, 2022, edition 3 (February 2022; corrected version March 2022); Accessed: 2026.05.25. [Online]. Available: https://www.iso.org/standard/75652.html

[21] American Institute of Certified Public Accountants (AICPA), "Trust services criteria (TSP section 100): 2017 with revised points of focus — 2022," AICPA Trust Services Criteria (basis for SOC 2 examinations), 2022.

[22] European Union, "Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the union (NIS2 directive)," Directive of the European Parliament and of the Council, 2022, oJ L 333, 27.12.2022; Accessed: 2026.05.25. [Online]. Available: https://eur-lex.europa.eu/eli/dir/2022/2555/oj

[23] National Institute of Standards and Technology (NIST), "The NIST cybersecurity framework (CSF) 2.0," NIST Cybersecurity White Paper (CSWP) 29, 2024, accessed: 2026.05.25. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf

[24] Center for Internet Security, "CIS critical security controls v8.1," Version 8.1, 6 2024, accessed: 2026.05.25. [Online]. Available: https://www.cisecurity.org/insights/white-papers/cis-critical-security-controls-v8-1

[25] X. Li, B. Liu, X. Bai, M. Zhang, Q. Zhang, Z. Li, H. Duan, and Q. Li, "Ghost domain reloaded: Vulnerable links in domain name delegation and revocation," in *Proceedings of the 2023 Network and Distributed System Security Symposium (NDSS '23)*. San Diego, CA, USA: The Internet Society, 2023.

[26] D. Liu and R. Duan, "Dangling domains: Security threats, detection and prevalence," 9 2021, accessed: 2026.05.25. [Online]. Available: https://unit42.paloaltonetworks.com/dangling-domains/

[27] M. Zorz, "BadDNS: Open-source tool checks for subdomain takeovers," 2 2025, accessed: 2026.05.25. [Online]. Available: https://www.helpnetsecurity.com/2025/02/03/baddns-open-source-tool-check-domain-subdomain-takeover/

[28] E. Borges, "DNSSEC: What is it? how does it work?" https://www.recordedfuture.com/threat-intelligence-101/tools-and-techniques/dnssec, 3 2024, accessed: 2026.05.25.

[29] SANS Internet Storm Center, "Quick status of the CAA DNS record adoption," https://isc.sans.edu/diary/26738, 10 2020, accessed: 2026.05.25.

[30] CISA, "DNS zone transfer AXFR requests may leak domain information," https://www.cisa.gov/news-events/alerts/2015/04/13/dns-zone-transfer-axfr-requests-may-leak-domain-information, 9 2016, accessed: 2026.05.25.

[31] Infoblox Threat Intel. (2024, 11) DNS Predators Hijack Domains to Supply their Attack Infrastructure. Accessed: 2026.05.25. [Online]. Available: https://blogs.infoblox.com/threat-intelligence/dns-predators-hijack-domains-to-supply-their-attack-infrastructure/

[32] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS security introduction and requirements," IETF RFC 4033 (Standards Track), 2005, accessed: 2026.05.25. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4033

[33] ——, "Resource records for the DNS security extensions," IETF RFC 4034 (Standards Track), 2005, accessed: 2026.05.25. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4034

[34] ——, "Protocol modifications for the DNS security extensions," IETF RFC 4035 (Standards Track), 2005, accessed: 2026.05.25. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4035

[35] J. Hoffman-Andrews, "DNS certification authority authorization (CAA) resource record," IETF RFC 8659 (Standards Track), 2019, accessed: 2026.05.25. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8659

[36] G. Akiwate, R. Sommese, M. Jonker, Z. Durumeric, K. C. Claffy, G. M. Voelker, and S. Savage, "Retroactive identification of targeted DNS infrastructure hijacking," in *Proceedings of the ACM Internet Measurement Conference (IMC '22)*. Nice, France: Association for Computing Machinery, 10 2022, pp. 14–32.

[37] A. Hern. (2015, 2) Lenovo website hacked and defaced by lizard squad in superfish protest. The Guardian. Accessed: 2026.05.25. [Online]. Available: https://www.theguardian.com/technology/2015/feb/26/lenovo-website-hacked-and-defaced-by-lizard-squad-in-superfish-protest

[38] CSC Global, "CSC's 2023 domain security report finds many global 2000 companies neglect their .ai domain extensions despite surge in popularity for artificial intelligence," 10 2023, accessed: 2026.05.25. [Online]. Available: https://web.archive.org/web/20231117032117/https://www.cscglobal.com/service/press/many-global-2000-companies-neglect-their-ai-domains/

[39] National Institute of Standards and Technology (NIST), "Digital identity guidelines," NIST Special Publication 800-63-4, 2025, accessed: 2026.05.25. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-4.pdf

[40] D. Chiba, H. Nakano, and T. Koide, "DomainDynamics: Advancing lifecycle-based risk assessment of domain names," *Computers & Security*, p. 104366, 2025, accessed: 2026.05.25. [Online]. Available: https://doi.org/10.1016/j.cose.2025.104366

[41] J. So, N. Miramirkhani, M. Ferdman, and N. Nikiforakis, "Domains do change their spots: Quantifying potential abuse of residual trust," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 2130–2144.

[42] B. Krebs, "That domain you forgot to renew? yeah, it's now stealing credit cards," *KrebsOnSecurity* (blog), 11 2018, accessed: 2026.05.25. [Online]. Available: https://krebsonsecurity.com/2018/11/that-domain-you-forgot-to-renew-yeah-its-now-stealing-credit-cards/

[43] R. Li, S. Xiao, B. Liu, Y. Lin, H. Duan, Q. Pan, J. Chen, J. Zhang, X. Liu, X. Lu, and J. Shao, "Bounce in the wild: A deep dive into email delivery failures from a large email service provider," in *Proceedings of the 2024 ACM on Internet Measurement Conference*, ser. IMC '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 659–673.

[44] The Hacker News, "Google OAuth vulnerability exposes millions via failed startup domains," accessed: 2026.05.25. [Online]. Available: https://thehackernews.com/2025/01/google-oauth-vulnerability-exposes.html

[45] J. Zirngibl, S. Deusch, P. Sattler, J. Aulbach, G. Carle, and M. Jonker, "Domain parking: Largely present, rarely considered!" in *Proceedings of the 6th Network Traffic Measurement and Analysis Conference (TMA '22)*. International Federation for Information Processing (IFIP), 2022, pp. 1–9, accessed: 2026.05.25. [Online]. Available: https://dl.ifip.org/db/conf/tma/tma2022/tma2022-paper26.pdf

[46] World Intellectual Property Organization (WIPO), "WIPO domain name report 2024: UDRP case filings remain strong," 2025, accessed: 2026.05.25. [Online]. Available: https://www.wipo.int/amc/en/domains/news/2025/news_0001.html

[47] J. Clayton. (2021, 4) Google Argentina's domain name bought by man for £2. Accessed: 2026.05.25. [Online]. Available: https://www.bbc.com/news/technology-56870270

[48] M. Marx, "Bogusbazaar: A criminal network of webshop fraudsters," Security Research Labs Blog, 5 2024, accessed: 2026.05.25. [Online]. Available: https://www.srlabs.de/blog-post/bogusbazaar

[49] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the ACM*, pp. 171–176, 3 1964.

[50] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Doklady Akademii Nauk SSSR*, pp. 845–848, 1965, accessed: 2026.05.25. [Online]. Available: https://www.mathnet.ru/eng/dan31411

[51] M. A. Jaro, "Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida," *Journal of the American Statistical Association*, pp. 414–420, 1989.

[52] W. E. Winkler, "String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage," in *Proceedings of the Section on Survey Research Methods*. American Statistical Association, 1990, pp. 354–359, accessed: 2026.05.25. [Online]. Available: https://files.eric.ed.gov/fulltext/ED325505.pdf

[53] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, pp. 379–423 and 623–656, 1948, accessed: 2026.05.25. [Online]. Available: https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Executive Cyber Risk Governance Using Integrated Key Performance and Key Risk Indicators: The BCCR-Card Framework

Alexander Lawall
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
e-mail: alexander.lawall@iu.org

Maik Drozdzynski
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
e-mail: maik.drozdzynski@iu.org

***Abstract*—Cybersecurity has evolved into a material enterprise risk, yet cybersecurity metrics are still predominantly managed at the operational level and remain weakly connected to executive performance steering and corporate risk governance. As a result, senior decision-makers often lack economically interpretable indicators that link cyber risk exposure to strategic objectives and financial outcomes. This paper presents the Balanced Chance and Cyber-Risk Card (BCCR-Card), a concept-oriented steering framework that integrates cybersecurity into Enterprise Performance Management (EPM). The contribution follows a conceptual design research approach, synthesizing established controlling frameworks, Balanced Scorecard (BSC) logic, and corporate risk management principles. Central to the framework is Cyber Loss Intensity (CLI), a normalized metric that relates quantified cyber loss exposure to operating performance and enables risk-adjusted executive steering. The BCCR-Card establishes an integrated Key Performance Indicators (KPI)/Key Risk Indicators (KRI) architecture that systematically translates technical cyber-risk signals into board-level, economically meaningful decision support.**

***Keywords*-Cyber Risk Management; Cybersecurity Metrics; Enterprise Perfomance Management; Key Risk Indicators (KRI); Key Performance Indicators (KPI); Cyber Loss Intensity (CLI); Executive Cybersecurity Governance; Balanced Scorecard (BSC); Steering Concept.**

I. Introduction

This work is an extended version of *The Balanced Chance & Cyber-Risk Card: Extending Reichmann's Multidimensional Controlling Framework for C-Level Steering in SMEs*, published at SECURWARE 2025 [1]. This paper generalizes and extends the approach toward integrating cybersecurity into Enterprise Performance Management (EPM) systems.

Over the past decade, the risk landscape for organizations across all industries has been fundamentally reshaped by cybercrime, making cyber risk one of the most consequential threats to corporate leadership. Alongside established risk classes such as regulatory and bureaucratic risk, geopolitical risk, and market risk, cybercrime is increasingly emerging as a dominant category that EPM and governance must address in a structured and measurable way.

Lawall and Drozdzynski highlighted in [1], drawing on multiple indicators including reporting by Germany's Federal Office for Information Security (BSI) and related sources, that the cyber threat environment in Germany has been rising steadily across industries and organization sizes [2]. When widening this perspective from a national to a global level, the same upward trajectory becomes evident, as the following figures demonstrate.

Recent official reporting illustrates both the scale and the acceleration of cyber-enabled threats across jurisdictions. In the United States, the FBI Internet Crime Complaint Center recorded 859,532 complaints in 2024 and reported losses of 16.6 billion US dollars, underlining that cyber-enabled fraud and extortion translate into materialized economic impacts at the national level [3].

In the United Kingdom, the National Cyber Security Centre (NCSC) received 1,727 incident indications, which were triaged into 429 incidents requiring support, of which 204 were classified as nationally significant and 18 as highly significant, indicating a growing share of severe incidents reaching national attention [4]. From a European threat landscape perspective, ENISA reports that its 2025 assessment analysed 4,875 incidents for the period from 1 July 2024 to 30 June 2025, reflecting the sustained frequency of impactful events in open source and member state shared reporting [5].

Comparable volume signals appear in national reporting systems: Switzerland's NCSC received 35,727 cyber incident reports in the first half of 2025, with 58 percent related to fraud, alongside 57 ransomware incident reports and 605 attempted CEO fraud cases [6].

In Australia, the Australian Signals Directorate reports more than 42,500 calls to the national cybersecurity hotline and responses to over 1,200 cybersecurity incidents in the financial year 2024 to 2025, and notes that its protective DNS blocked access to 334 million malicious domains [7].

In Singapore, the Cyber Security Agency reported 159 ransomware cases in 2024 and observed 117,300 infected systems, indicating how ransomware and widespread commodity infections remain persistent across sectors [8]. Finally, a malware centric indicator from Germany's Federal Office for Information Security (BSI) reports around 6.50 million new malware variants in November 2025, averaging 217,000 new variants per day, illustrating the continuous industrialisation of malicious tooling [9]. Collectively, these official statistics support the conclusion that cyber risk is no longer a niche operational concern but a pervasive, high-velocity risk class that must be integrated into executive steering and EPM. This conclusion is strongly reinforced by Germany's Federal Office for Information Security (BSI), which states in their most recent annual report that "the analysis and management of

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

an organization's attack surfaces must be understood in every company and every institution, whether small or large, as an indispensable part of effective risk management" [2, p. 3].

As a consequence, cybersecurity should be a mandatory part in the boardroom agenda of company leaders. Controlling researcher Horváth lists cyber risk management, alongside Environmental, Social and Corporate Governance (ESG) reporting, among the fastest-growing controlling disciplines for CFOs in 2024 [10]. Yet existing research still lacks an integrated steering framework that treats cyber risks on an equal footing with classical corporate-risk categories. The Balanced Chance and Risk Card proposed in 2000 [11] and updated alongside the Law on Control and Transparency in the Corporate Sector (KonTraG) in 2001 [12, pp. 282] by controlling pioneer Reichmann offers a conceptual anchor as a breakthrough in risk-management, but has so far not been extended with cyber-specific perspectives. Likewise, the risk-controlling process by risk-management researcher Diederichs provides a systemic approach and does not yet incorporate the distinctive dynamics of cyber-threat scenarios [13, pp. 189].

Against this backdrop of a rapidly intensifying threat environment, many organizations have already elevated cybersecurity to the executive level by establishing dedicated C level accountability, for example through the role of the Chief Information Security Officer (CISO). This institutional anchoring is an important prerequisite, yet it does not automatically translate into an integrated steering logic. In practice, cybersecurity metrics are often managed in parallel to EPM and remain weakly connected to concept oriented analysis systems such as the Balanced Scorecard (BSC), to coherent Key Performance Indicator (KPI) systems, and to Key Risk Indicators (KRI) that are explicitly aligned with financial and operational steering objectives. As a result, cybersecurity and cyber resilience frequently lack a shared measurement and decision framework that links technical exposure and control effectiveness to enterprise wide performance targets and risk appetite.

This governance relevance is further reinforced by emerging European cybersecurity regulation. The NIS2 Directive significantly expands cybersecurity governance obligations for operators of essential and important entities, including mandatory risk-management measures, incident-reporting obligations, and explicit management accountability for cybersecurity oversight [14]. In parallel, the European Cyber Resilience Act (CRA) introduces security-by-design and vulnerability-management obligations for manufacturers of digital products across the entire product lifecycle [15]. Collectively, these regulations increase the need for integrated executive-level cyber-risk steering mechanisms that connect technical cybersecurity exposure with organizational governance, operational resilience, and economic performance management.

To address this gap, this paper generalizes the approach toward integrating cybersecurity into concept-oriented EPM systems by combining (i) EPM and controlling as the overarching steering process, (ii) concept-oriented analysis systems as the structuring logic for objectives and cause–effect relations, (iii) KPI and KRI systems as the operational measurement layer, and (iv) corporate risk management as the governance logic for identification, assessment, treatment, monitoring, and escalation of risks. The remainder of this paper is structured as follows:

Section II develops the theoretical foundation for these building blocks. Section III introduces the integrated concept and its design rationale. Section IV discusses implementation considerations and limitations. The paper then discusses generalizability and lessons learned within Section V. The final Section VI concludes the paper and outlines directions for future work.

II. Foundations of Enterprise Performance and Cyber Risk Management

A. Enterprise Performance Management and Controlling

EPM is commonly framed as an integrated management process that translates strategy into measurable objectives, aligns planning and resource allocation, and steers organizational performance through consistent measurement, analysis, and decision routines. In this sense, EPM is not merely a reporting activity but a coordinated set of managerial practices that connects strategic intent with operational execution and enables performance-oriented governance across the enterprise. A central idea in the EPM literature is the purposeful integration of complementary management methods, such as strategy mapping, BSC logic, and coherent performance measures, into a unified steering approach that improves decision-making and organizational control [16].

This process-oriented understanding of EPM is closely related to the Controlling tradition. Controlling conceptualizations emphasize the design and use of integrated information and steering systems to support planning, control, and management decision-making [17]. In particular, the controlling perspective highlights the necessity of structuring objectives, indicators, and responsibilities into an internally consistent steering system and of embedding measurement into recurring management cycles [18]. From this viewpoint, EPM can be interpreted as an internationally established umbrella term for integrated steering, while Controlling provides well-developed conceptual foundations for designing measurement-based governance and for clarifying roles and responsibilities in performance-oriented management [19].

B. Concept-Oriented Analysis Systems

Concept-oriented analysis systems are management support approaches that deliberately operationalize a specific managerial or business concept and, by doing so, impose a structured logic on data selection, indicator definitions, and interpretation. Rather than offering only generic reporting or ad hoc analysis, they build on the functionalities of underlying analytical base systems and translate an established business framework, such as the BSC, value-based management, or risk management, into a coherent analysis and steering logic [18]. Consequently, concept-oriented analysis systems are not intended to be used in an arbitrary or opportunistic manner. The following

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

subsections illustrate this logic with two prominent concept-oriented approaches. They also motivate why a dedicated KPI and KRI layer is required to operationalize these concepts in a measurable and governance-ready manner, which is addressed in subsection II-E.

1) Reichmann's Multidimensional Controlling Framework: The multidimensional controlling concept developed by Thomas Reichmann is considered a reference model in the field of controlling, because it integrates functional responsibilities, information logic, and time horizons within a single, coherent framework [18, pp. 21]. At its core, controlling is defined as an IT-supported, decision-oriented management service: every decision-maker should receive exactly the information that matches their task, planning horizon, and area of responsibility.

The model is built around dimensions. (i) Functional view (e.g., cost- and profit-, financial, procurement, or risk controlling) allocates information along the value chain and thus provides an impact-oriented perspective. (ii) Information categories separate monetary profit- and cash-flow figures from operational quantity and quality data, enabling quantitative metrics to be combined with qualitative early-warning indicators. (iii) Time horizons distinguish strategic, tactical, and operational scopes. consequently, short-term variance analyses and long-term trend observations can coexist within the same steering model.

It is practically relevant due to its integration blueprint, where each dimension assumes a distinct role in the IT architecture. Fact and dimension tables in a data warehouse map functions, information categories, and time horizons. Extract, Transformation and Load (ETL)-processes transport booking data up to the key-figure layer and dashboards. Planning, actual, and forecast values can therefore be compared across all levels without media or aggregation breaks. In practice, boards decide based on top-level KPIs (Return on Investment, working-capital ratio, etc.), while divisional managers drill down to variance reports. Meanwhile, operational controllers still work with itemized lists.

Finally, the concept supports early-warning and scenario analyses, so that qualitative indicators (e.g., Threats or market-trend signals) are stored as a distinct information category and can be combined with monetary KPIs. Organisations thus detect opportunities and risks earlier and can simulate response options before effects appear in the income statement.

In summary, Reichmann's multidimensional concept provides a robust bridge between a company's goal system and its technical implementation, allowing for traceable aggregation from primary data to key figures and providing role-specific access to the exact level of information granularity required for sound management decisions.

Reichmann's merit lies not only in having proposed a controlling framework in the mid-1980s, but also in designing it to stay compatible with future technologies and thus continuously extensible. Although conceived decades ago, the model is regularly adapted to new industries and technologies, allowing emerging controlling sub-disciplines, such as risk controlling, to be integrated without altering its core. For example, Drozdzynski embeds medical performance data and BI dashboards into the system- and application-layers of Reichmann's concept for a hospital context [20], while Liebe and Drozdzynski extend the framework to health-and-social-care organisations [21]. These adaptations demonstrate that the framework remains comparable across sectors, whereas its special part can be customised with domain-specific metrics, especially for cybersecurity.

2) The Balanced Scorecard (Kaplan and Norton): The BSC is a prominent concept-oriented analysis system that translates strategy into a coherent set of performance objectives and measures. Kaplan and Norton originally introduced the BSC as a framework that complements traditional financial measures with nonfinancial perspectives, thereby enabling management to monitor current results while simultaneously steering the capabilities and drivers required for future performance [22]. Subsequent work positioned the BSC explicitly as a strategic management system that links objectives, measures, targets, and initiatives and embeds measurement into recurring management processes, including communication, alignment, and learning [23]. In this understanding, the BSC is not a generic dashboard but a structured steering logic in which indicators are selected and interpreted based on their role within a strategy model, often operationalized through cause effect relations and strategy maps [24].

Over three decades of research and practice, the BSC has evolved into a highly adaptable framework that is frequently extended to address new strategic domains and stakeholder expectations, while retaining its core idea of balancing multiple perspectives in a unified management logic. Recent systematic reviews summarize this development and highlight trends such as domain-specific scorecards, sustainability-oriented BSCs, and the increased emphasis on governance and strategic alignment [25], [26]. In the cybersecurity domain, research has proposed information security BSC variants that integrate security governance requirements with structured performance measurement for senior management, for example by mapping established control and governance frameworks to a BSC logic [27]. This makes the BSC a natural conceptual bridge for integrating cybersecurity objectives and indicators into EPM, because it offers an established mechanism to connect strategic intent with operational measurement and management decision routines across the organization [23], [25].

C. Corporate Risk Controlling

In managerial accounting, risk management refers to the systematic handling of uncertainties that may impair, or enhance, the achievement of corporate objectives [13]. From an expected-value perspective, risk is the dispersion of potential outcomes around a planned value [11]. Hopkin further argues that modern frameworks must recognize the upside of uncertainty and integrate opportunity management into corporate steering [28, p.472].

The international guidelines refine the process. ISO 31000:2018 embeds risk management within governance

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

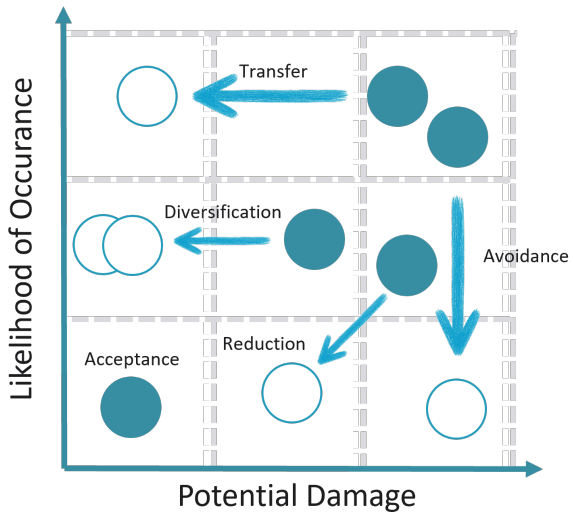


Figure 1. Risk Map for Likelihood–Impact-Based Corporate Risk Steering

structures, while Committee of Sponsoring Organizations of the Treadway Commission Enterprise Risk Management (COSO ERM) 2017 operationalises a four-step cycle of (i) Identify, (ii) Assess, (iii) Respond, and (iv) Monitor [29]. Following this tradition, Diederichs draws a clear line between risk management (strategic orientation) and risk controlling (information supply and steering). Risk controlling comprises (i) quantitative appraisal through scenario and sensitivity analyses, (ii) portfolio aggregation into metrics, such as Value at Risk, and (iii) stakeholder-specific reporting to boards and operational units [13].

Figure 1 illustrates a typical risk map used in risk controlling to prioritize risks and select normative risk treatment strategies. The vertical axis represents the likelihood of occurrence, while the horizontal axis represents the magnitude or impact of the loss. In cybersecurity contexts, the likelihood assessment is typically derived from a combination of exposure-related and adversary-related factors. Exposure-related factors include the accessibility and criticality of attack surfaces, such as internet-facing systems, remote-access interfaces, cloud dependencies, or supply-chain connectivity. Adversary-related factors include the required level of attacker sophistication, availability of exploit tooling, existence of known vulnerabilities, and the expected effort required to execute a successful attack. Consequently, likelihood determination is not interpreted as a purely statistical probability, but rather as a structured managerial estimation informed by technical exposure, threat intelligence, and organizational context.

The dashed grid partitions the space into discrete risk classes, for example, low, medium, and high, which supports management communication and enables the definition of explicit thresholds for escalation. Each marker represents a risk position. In the diagram's visual logic, filled markers represent the current, assessed risk exposure, whereas hollow markers indicate the desired target position or the residual risk after the implementation of treatment measures. The arrows depict steering intentions. Moving a risk from its current cell toward a target cell that better aligns with the organization's risk appetite and tolerance.

The map also embeds the main normative risk treatment strategies. Risk acceptance is associated with the low-likelihood, low-impact region. In risk controlling, acceptance does not mean ignoring the risk, but rather an explicit decision to tolerate it under defined conditions, combined with monitoring and predefined triggers that initiate treatment if the risk drifts into a higher zone. Risk reduction (or mitigation) is visualized as a diagonal movement down and to the left, corresponding to decreasing both likelihood and potential damage. Operationally, this strategy is realized through preventive and detective controls that reduce the likelihood of occurrence, complemented by resilience measures that reduce impact if the event materializes, such as containment capabilities, backup and recovery, and incident response readiness.

Risk avoidance is depicted as a movement toward substantially lower likelihood and or lower impact, reflecting the elimination of the risk source by discontinuing the activity, changing the process, or redesigning system boundaries so that the adverse event can no longer occur in its original form. Risk transfer (or risk shifting) is primarily shown as a shift toward lower impact at the same likelihood. Conceptually, transfer means that the event may still occur, but the financial consequences are contractually shifted or shared with third parties, for example, through insurance, outsourcing arrangements with defined liabilities, or other risk-sharing mechanisms.

Risk diversification is presented as a portfolio-oriented strategy. Instead of relying on a single measure, exposure is spread across independent assets, suppliers, technologies, or locations to reduce concentration risk and limit worst-case losses. On the map, this is reflected by multiple target markers in a lower-impact region, emphasizing that diversification aims to reduce the severity of aggregated outcomes even if individual adverse events remain possible.

From a risk controlling perspective, the value of this representation is that it connects assessment to steering. The map makes transparent which risks exceed tolerable zones, which treatment strategy is chosen for each class, and what residual position management intends to achieve. It also supports governance by clarifying responsibilities, defining escalation rules by zone, and enabling a structured comparison of expected risk movement against the cost and feasibility of measures.

From an operator perspective, the resulting risk position additionally depends on the deployment and operational environment in which systems and processes are embedded. Identical technical vulnerabilities may produce substantially different risk exposures depending on factors such as operational criticality, degree of network segmentation, availability requirements, regulatory obligations, interdependencies with third parties, and the organization's recovery capabilities. Therefore, risk assessment within the proposed steering framework must always be interpreted in relation to the specific

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

operational context.

D. Balanced Chance and Risk Card

The Balanced Chance and Risk Card (BCRC) was introduced by Reichmann as an extension of the BSC to meet the tighterregulatory requirements for integrated risk management, such as KonTraG in Germany, at the beginning of the 2000s [22] [12]. Diederichs subsequently operationalised the concept for controlling practice and anchored it in the German "Controlling" journal [30]. The instrument combines value-based management, the BSC logic, and systematic opportunity-and-risk control within a single reporting artefact.

Several authors recommend a six-step implementation procedure: (i) define strategic goals per perspective, (ii) derive appropriate performance KPIs, (iii) identify and evaluate the main opportunities and risks (likelihood $\times$ impact), (iv) link KPIs with the respective opportunities/risks to obtain risk-adjusted targets and actuals, (v) specify measures, owners and milestones, and (vi) install a rolling review cycle (monthly or quarterly). This procedure merges strategy progress, risk exposure, and action effectiveness into a single management view.

While the BCRC is conceived as an entirely risk-oriented steering framework, recent applications mention cyber threats only in passing as a subset of generic operational risks and provide neither dedicated KRIs nor tailored control routines for them [30]. Considering the accelerating frequency, networked propagation and potentially existential financial impact of contemporary cyber incidents, it is timely and methodologically warranted to give cybersecurity risks disproportionate analytical weight within the BCRC [31]. Section II-F therefore examines the nature and managerial relevance of cybersecurity risks as a prerequisite for their systematic integration into the card.

E. Key Performance Indicators and Key Risk Indicators

KPIs are a subset of performance measures that are deliberately selected because they represent the most decision-relevant aspects of organizational performance in relation to the company's objectives. In the research literature, the terms performance indicator, performance measure, performance metric, and KPI are often used interchangeably. In this paper, the term KPI refers to a carefully selected set of indicators with the specific managerial purpose to operationalize strategy, support steering decisions, and structure performance communication within recurring management cycles [32]. In this process-oriented understanding, KPIs serve as the measurement layer of EPM. With that said, they make the achievement of objectives observable, enable target setting and variance analysis, and provide a basis for management interventions when performance deviates from expectations [16].

From a controlling perspective, the usefulness of KPIs depends less on the availability of data and more on conceptual fit and design quality. KPIs should therefore be derived from the organization's objectives and value drivers rather than being opportunistically chosen from existing reports [17]. Moreover, KPIs require an operational definition that ensures consistent calculation and interpretation over time and across organizational units, including explicit specification of scope, unit of measure, data origin, and calculation rules [18]. KPIs should also be actionable and at least partially controllable by responsible actors, so that deviations trigger meaningful responses rather than only retrospective explanations [33]. Finally, KPI governance matters in a way that ownership, reporting cadence, and target setting must be aligned with decision processes, because indicators that are not embedded in decision routines tend to degenerate into passive reporting and do not contribute to steering [17], [33].

A brief illustrative example clarifies the intended abstraction level without anticipating the implementation section. In an availability-sensitive context, service availability can be defined as a KPI because it directly reflects a strategic performance promise, can be measured consistently, and supports clear management actions when targets are missed. The essential point is that KPIs translate strategic intent into a manageable set of operational measurable signals that connect performance monitoring with managerial decision making, thereby enabling EPM and controlling to operate as an integrated steering process [16], [17].

KRIs are metrics designed to provide early warning signals about changes in a company's risk exposure, risk drivers, or control environment, thereby supporting proactive risk management decisions. Sound practice guidance emphasizes that KRIs should be linked to specific risks, defined with thresholds and escalation rules, and embedded into monitoring and reporting routines so that they trigger timely management attention before risk events materialize [34]. In contrast to KPIs, which primarily measure the achievement of objectives and performance outcomes, KRIs are primarily oriented toward risk appetite and risk tolerance, focusing on the likelihood and or impact of adverse events and the conditions under which these risks may increase [34], [35].

Despite this distinction, KPIs and KRIs are conceptually coupled within integrated steering. Performance objectives and the KPIs used to manage them imply an underlying risk profile, because pursuing performance targets often increases exposure to certain threats, dependencies, and control trade offs. Accordingly, a coherent steering system aligns KPIs with a complementary set of KRIs that monitor risk drivers that could undermine KPI target achievement [35]. For instance, while a KPI may reflect on time delivery or system availability, a KRI may track the exposure drivers that threaten these outcomes, such as critical vulnerability backlog or phishing susceptibility, thereby enabling management to intervene before performance degradation occurs. This coupling is central for integrating cybersecurity into EPM, because cybersecurity metrics often function as risk driver indicators and control effectiveness indicators, which need to be linked to enterprise level performance objectives through a clear governance logic in corporate risk management [17], [34].

Finally, the distinction between KPIs and KRIs is not purely semantic but affects design and governance. KPIs are typically

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

evaluated against performance targets and incentives, whereas KRIs require risk thresholds, escalation paths, and explicit interpretation within risk appetite statements [34]. Treating KRIs as if they were KPIs can create dysfunctional behavior, for example optimizing the reported indicator without addressing the underlying risk driver. Conversely, treating KPIs without considering their risk drivers can lead to performance steering that systematically increases unmanaged exposure.

Therefore, an integrated measurement layer should define both KPI and KRI roles explicitly and embed them into concept-oriented analysis systems and the corporate risk management cycle, which provides the decision rules for acceptance, mitigation, transfer, or avoidance.

F. Cyber Risk and Cybersecurity Threat Dynamics

Recent research highlights the importance of integrating real-time Cyber Threat Intelligence into dynamic risk management systems to enable situational awareness [36]. A semantic web technology-based architecture is introduced to create a dynamic risk assessment system at operational, tactical, and strategic levels [37]. A further development is the concept with an ontology-driven real-time risk management approach that encompasses anomaly detection and cataloging vulnerabilities [38]. The requirement for automated technologies to offer situational awareness solutions for National Cyber Operation Centers is pointed out by [39]. An example in practice suggests a Metrics Visualization System that can dynamically visualize network security incidents and correlate them with risk levels [40]. Collectively, these studies emphasize the potential for real-time, standardized operational cyber threat metrics to enhance decision-making across hierarchy levels, from administrators to the C-suite, through a more timely and accurate assessment of an organization's cybersecurity position.

CTI has proved to be an essential way of supplementing cybersecurity and risk management in organizations. CTI significantly increases threat detection, response, and risk management capability [41]. CTI provides evidence-based insight into the threats to facilitate proactive risk mitigation in critical infrastructure [42]. CTI can be integrated into campaigns for raising awareness against cyberattacks, especially in the banking sector, through the use of tactical, operational, and strategic intelligence [43] [36]. In response to the need for real-time risk analysis, a semantic-based architecture using Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL) has been proposed, such as Structured Threat Information eXpression (STIX) v2.0 for the structured exchange of threat information [37]. Despite its advantages, there are barriers to CTI adoption, including technological constraints and the absence of executive sponsorship. These issues need to be overcome through extensive awareness programs, executive participation, and systematic training efforts [41].

III. The BCCR-Card: A Cyber Risk-integrated Steering Concept

A. Limitations of Classical Risk Assessment Models in Cyber Contexts

Classical risk-assessment frameworks have challenges in cyber domains for four core reasons. First, scarce loss data leave actuarial or scenario models without reliable frequency and severity inputs [44]. Second, traffic-light heat maps compress complex threats into ordinal colours, masking value at stake and skewing priorities [45]. Third, adversarial tactics evolve weekly, so annual risk registers are inappropriate, as European Union Agency for Cybersecurity (ENISA) 2024 survey warns [46]. Fourth, cloud and supply-chain interdependence creates cascade-prone losses; single-asset Value at Risk (VaR) thus understates extremes [47]. To close the gaps, this paper introduces the BCCR-Card as a steering concept that combines traditional financial and qualitative steering practices with modern cybersecurity requirements.

Figure 2 shows the target architecture of a cybersecurity-integrated steering concept that synthesizes an economic, concept-oriented management view with a technical cybersecurity measurement view. In the upper part, two input domains are shown as separate but complementary sources of steering logic. On the left, "Concept-oriented Steering" represents the economic view and bundles established controlling components, namely a controlling framework, a BSC logic, and a KPI layer, complemented by risk management as the governance mechanism for identifying, prioritizing, and treating risks. On the right, "Cybersecurity" represents the technical view and contributes a dedicated KRI layer, i.e., measurable signals that capture changes in cybersecurity exposure, threat dynamics, and control effectiveness. Both domains feed into the integrated steering concept depicted in the lower part of the figure.

The lower part operationalizes the synthesis as an extended scorecard-like structure with five perspectives: Finance, Processes, Market, Employees, and an added Cyber Resilience perspective. This extension reflects the core design intent that cybersecurity is not managed as an isolated technical silo but becomes an explicit steering dimension alongside established enterprise perspectives. The vertical "Risk Management" element and the risk map on the left indicate that the risk management process serves as a trigger and prioritization mechanism: risks with high impact and likelihood are routed into the steering concept and must be systematically addressed through defined measures. In the integrated concept, these measures are represented by a consistent indicator pair consisting of KPIs and KRIs: KPIs reflect the progress and effectiveness of managerial actions from the enterprise perspective, while KRIs provide early-warning and risk-driver signals that justify escalation and adaptive steering. The dashed red arrows illustrate this coupling by showing how a cyber risk signal (exemplified by a KRI in the Cyber Resilience perspective) propagates into performance steering across the other perspectives, ensuring that cybersecurity risks are translated into enterprise-relevant

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

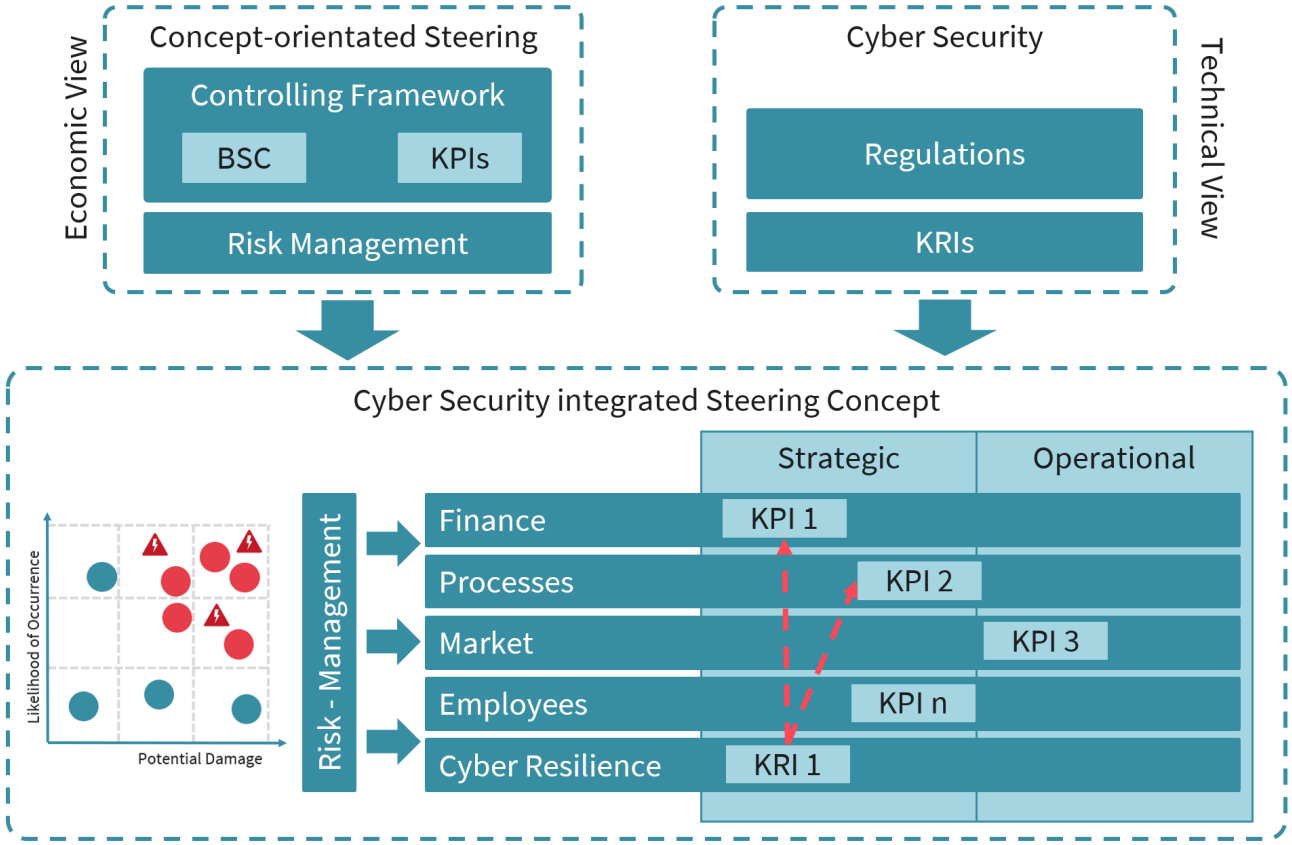


Figure 2. Integrated Enterprise Performance and Cyber Risk Steering Architecture

management actions and performance objectives rather than remaining confined to technical reporting.

Finally, the figure differentiates between an operational and a strategic layer across the scorecard. This indicates a two-level steering logic: operational indicators support short-cycle monitoring and control (e.g., execution and effectiveness of measures), whereas strategic indicators support long-term capability building, resilience objectives, and alignment with enterprise strategy. Overall, the target image communicates a governance-ready integration approach in which every prioritized cyber risk is linked to accountable management actions and measurable outcomes through a coherent KPI/KRI system embedded within a concept-oriented EPM structure.

B. Structured Integration of Cyber Risk into Enterprise Steering

Modern organizations face data overload and goal conflicts. An integrated controlling concept mitigates both by (i) aligning metrics with strategic objectives, (ii) enforcing a common language for financial and non-financial data, and (iii) enabling transparent, audit-ready decision trails [48, pp. 7] [49, pp. 5]. Research shows that companies with coherent management-control systems achieve higher decision quality and risk resilience than those using ad-hoc indicator sets [50, pp. 30].

Applying this logic to cybersecurity avoids metric issues: isolated dashboards might track patch rates or incident counts, yet without linkage to profitability and liquidity, they lack managerial traction. Embedding cyber-risk KPIs into Reichmann's cube, e.g., as an additional information category on the system layer, ensures goal congruence (security spend vs. value at risk), comparability (cross-unit benchmarking), and governance compliance (StaRUG early-warning duties). Hence, a structured concept is not academic ornamentation but a prerequisite for turning raw cyber data into actionable, strategy-consistent steering information.

The proposed framework introduces cyber resilience as a fifth dimension besides the well-known four dimensions: finance, growth, internal processes, and customer/market. Clear roles and responsibilities ensure accountability, like the Chief Executive Officer (CEO)/Chief Financial Officer(CFO) owns capital allocation and is in charge of gaining profitability and driving financial return. The top management, e.g., Vice President Sales (VPS), owns the Market/Customer perspective. The Chief Information Security Officer (CISO) operates the technical control loop and supplies metrics alongside the perspective *cyber resilience*. The following concept will not discuss the steering capabilities of the BSC in general, but will focus on the steering levers in the field of cyber risk management.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

C. Development of Cyber-Relevant KPIs and KRIs

This section operationalizes the integrated steering concept by outlining exemplary KPIs and KRIs for each scorecard perspective (Finance, Growth, Market, Process, and Cyber Resilience). The indicators are intentionally presented at an abstract level to illustrate the measurement logic and the intended coupling of performance steering and risk steering. Concrete target values, thresholds, and data-source-specific implementation details depend on the company's size, branch, and business model. In line with established guidance on risk indicators, the proposed KRIs are designed as early-warning signals that can be linked to explicit thresholds and escalation rules within the corporate risk management cycle.

In the context of this paper, cyber resilience is understood as the organizational capability to maintain acceptable operational functionality during cyber incidents while simultaneously ensuring rapid recovery and restoration after disruption. This definition deliberately combines two complementary dimensions frequently discussed in resilience research: (i) resistance and continuity capability under attack, including degradation-tolerant operation and containment, and (ii) recovery capability, including restoration speed, adaptability, and learning after incidents. Both dimensions are within the scope of the BCCR-Card because they influence enterprise performance differently. Resistance-oriented capabilities primarily affect service continuity and operational stability, whereas recovery-oriented capabilities primarily influence incident duration, financial loss severity, and long-term organizational resilience.

1) Cross-perspective top indicator: Cyber Loss Intensity (CLI): The most central indicator in the proposed system is the CLI, which explicitly links economically interpretable cyber-risk exposure to a core performance result. CLI is defined as the ratio of the organization's Expected Annual Loss (EAL) from cyber incidents to the financial success EBIT:

$$\mathrm{CLI} = \frac{\mathrm{EAL\ (Cyber)}}{\mathrm{EBIT}} \tag{1}$$

EAL represents the anticipated annualized financial impact of cyber incidents, aggregating the expected values of relevant cyber-loss scenarios across the organization. It can be operationalized through scenario-based quantification, combining estimates of incident frequency with estimates of loss severity, and is commonly used in risk quantification contexts to make cyber risk economically interpretable. The EBIT serves as a widely used measure of operational performance and is therefore an appropriate denominator for expressing the intensity of cyber-loss exposure relative to operating earnings. By construction, CLI expresses the share of operating performance at risk from cyber incidents on an expected-value basis. A rising CLI indicates that cyber risk increasingly threatens the organization's ability to realize operating results, thus providing a board-level signal for prioritization, resource allocation, and risk treatment decisions. In contrast to absolute loss estimates, CLI also improves comparability across organizations and over time by normalizing cyber exposure to a firm-specific performance scale. From a steering perspective, CLI is intentionally cross-perspective: it is grounded in cyber-risk quantification (Cyber Resilience perspective) while being interpreted as performance risk within the Finance perspective. This makes CLI a core bridge between cyber resilience management and EPM.

2) Finance perspective: The Finance perspective retains conventional performance KPIs that reflect value creation and financial sustainability, such as Return on Equity (ROE). ROE captures profitability relative to shareholders' equity and provides a high-level signal for steering financial performance. In the BCCR-Card, ROE is complemented by the CLI-provided cyber-performance linkage. While ROE indicates realized financial performance, CLI expresses forward-looking loss exposure relative to operating earnings and thereby supports risk-adjusted performance steering.

a) Growth perspective: The Growth perspective captures organizational capabilities that enable long-term performance, including human capital, learning, and cultural drivers. Exemplary KPIs include Employee Satisfaction and Training Hours per Year. Employee Satisfaction functions as a proxy for engagement and retention risks and is frequently linked to productivity and capability building. Training Hours per Year operationalizes investment into skills and organizational learning. In the cybersecurity-integrated context, these KPIs are complemented by a human-factor KRI, the Phishing Click-Through Rate, which measures susceptibility to social engineering. A higher click-through rate indicates a higher likelihood of compromise and, therefore, increased cyber-risk exposure. As an early-warning indicator, it is suitable for threshold-based escalation and for steering preventive measures such as awareness programs and secure-by-design process changes.

b) Market perspective: The Market perspective represents customer and market outcomes. Exemplary KPIs include Net Promoter Score (NPS) and Customer Retention Rate. NPS captures perceived customer value and loyalty tendencies, while retention indicates the ability to maintain customer relationships over time. From a cybersecurity-integrated perspective, these KPIs are complemented by Service Downtime per Customer, which captures the customer-facing impact of cyber incidents and resilience deficits. Downtime per customer serves as an impact-oriented indicator, connecting technical disruption to customer experience and potential churn, thereby providing a measurable link between cyber incidents and market outcomes.

c) Process perspective: The Process perspective reflects operational execution and process quality. An exemplary KPI is the Warranty Rate, which indicates process-related quality outcomes and associated costs. In the cybersecurity-integrated logic, the Process perspective also includes control-effectiveness and exposure-driver indicators such as Patch Compliance Rate. Patch Compliance Rate measures the share of systems that meet defined patching requirements within a given time window. It functions as a leading indicator for technical exposure because patching discipline directly

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

influences vulnerability windows and exploitability. As a KRI, it supports proactive steering by enabling management to intervene before vulnerabilities translate into incidents [34].

d) Cyber Resilience perspective: The Cyber Resilience perspective consolidates both economic cyber-risk quantification and operational resilience capability. Key indicators include EAL and Cyber Value at Risk (CyVaR) as economic measures, and Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR) as operational resilience measures. EAL provides an expected-value estimate of annual cyber-loss exposure, while CyVaR expresses the potential loss magnitude at a defined confidence level, capturing tail risk beyond expected values [44]. MTTD and MTTR measure detection and response performance, i.e., the organization's capability to identify and contain incidents quickly, which is critical for limiting realized impact. A Cyber Resilience Index can serve as an aggregated management indicator that summarizes multiple resilience dimensions for executive reporting, while maintaining drill-down capability into the underlying component indicators when steering decisions require diagnostic detail.

Overall, the proposed indicator set demonstrates how cyber-risk drivers, control effectiveness, operational resilience, and economic impact measures can be linked to conventional enterprise performance steering within a concept-oriented scorecard. The CLI plays a central role in this architecture by translating quantified cyber-loss exposure into a performance-relevant intensity measure that is directly interpretable by executive management and controlling.

Furthermore, distinguishing between *gross (inherent)* and *residual (net)* cyber risk is essential. This enables management to assess the effectiveness of existing controls. For example, if the inherent ransomware risk is high, but the residual risk is low due to segmentation and offline backups, no immediate investment is needed. German legal standards under KonTraG and StaRUG explicitly require this level of risk quantification [51] [52].

In summary, embedding cyber KPIs into a concept-oriented steering system bridges technical telemetry and economic EPM. For companies, this approach is not only methodologically sound but regulatory-aligned, promoting a risk-aware leadership culture with measurable security accountability.

D. The BCCR-Card in Practice

Figure 3 operationalizes the abstract steering architecture introduced in Figure 2. While Figure 2 illustrates the conceptual integration of risk management, KPI/KRI coupling, and cyber resilience within an enterprise steering architecture, Figure 3 translates this logic into an exemplary scorecard instantiation with perspective-specific indicators, organizational ownership, and management-level accountability. In this sense, the Cyber Resilience perspective shown in Figure 3 represents the practical realization of the resilience-oriented steering layer introduced conceptually in Figure 2.

Figure 3 illustrates a BCCR-Card that extends the classical BSC logic by introducing an explicit Cyber Resilience perspective and by integrating cyber-related risk indicators with conventional performance indicators. The scorecard is structured as introduced earlier, thereby positioning cybersecurity and resilience as a dedicated management dimension alongside established enterprise perspectives rather than treating them as an isolated technical domain.

A defining characteristic of the BCCR-Card is its integrated KPI/KRI architecture, which makes cyber risk measurable and economically interpretable within regular management steering routines. The grey-blue elements represent performance-oriented KPIs used to monitor the achievement of strategic objectives in each perspective. From a finance perspective, this includes conventional profitability measures, such as Return on Equity, complemented by finance-related cyber impact measures, such as CLI, that make cyber-induced financial exposure visible at the executive level. The growth perspective combines organizational capability and learning signals, exemplified by employee satisfaction and training hours per year, and augments them with a human-factor risk driver indicator, such as the phishing click-through Rate, which operationalizes security culture and vulnerability to social engineering. The Market perspective links customer and market outcomes, for example, through net promoter score and customer retention rate, and explicitly accounts for cyber-related service degradation through indicators such as service downtime per customer, reflecting that cyber events can directly affect customer experience, reputation, and churn. The process perspective contains operational execution measures, such as Warranty Rate, and introduces cyber control effectiveness signals, such as patch compliance rate, which serves as a leading indicator of technical exposure because patching discipline directly affects vulnerability windows and exploitability.

The Cyber Resilience perspective consolidates risk and resilience measures that connect technical security posture to managerial and financial steering. It includes economically interpretable risk measures such as EAL and CyVaR, which express cyber exposure in monetary terms and support risk appetite discussions and resource allocation. In addition, it includes operational resilience measures such as MTTD and MTTR, which operationalize detection and response capability and therefore the organization's ability to limit impact once an incident occurs. An aggregated Cyber Resilience Index can serve as a synthetic indicator to communicate resilience maturity and trend development to senior management while maintaining the possibility of drilling down into contributing sub-metrics when required.

The dashed red arrows emphasize the cause–effect logic underpinning the scorecard. They illustrate how cyber risk drivers and deficiencies in control effectiveness can propagate beyond the Cyber Resilience perspective and affect performance outcomes in other perspectives—for example, how weak patch compliance or elevated phishing susceptibility may increase incident likelihood, trigger outages, deteriorate customer experience, and eventually translate into financial impacts. Finally, the figure conveys a governance and accountability concept through organizational layering and role

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

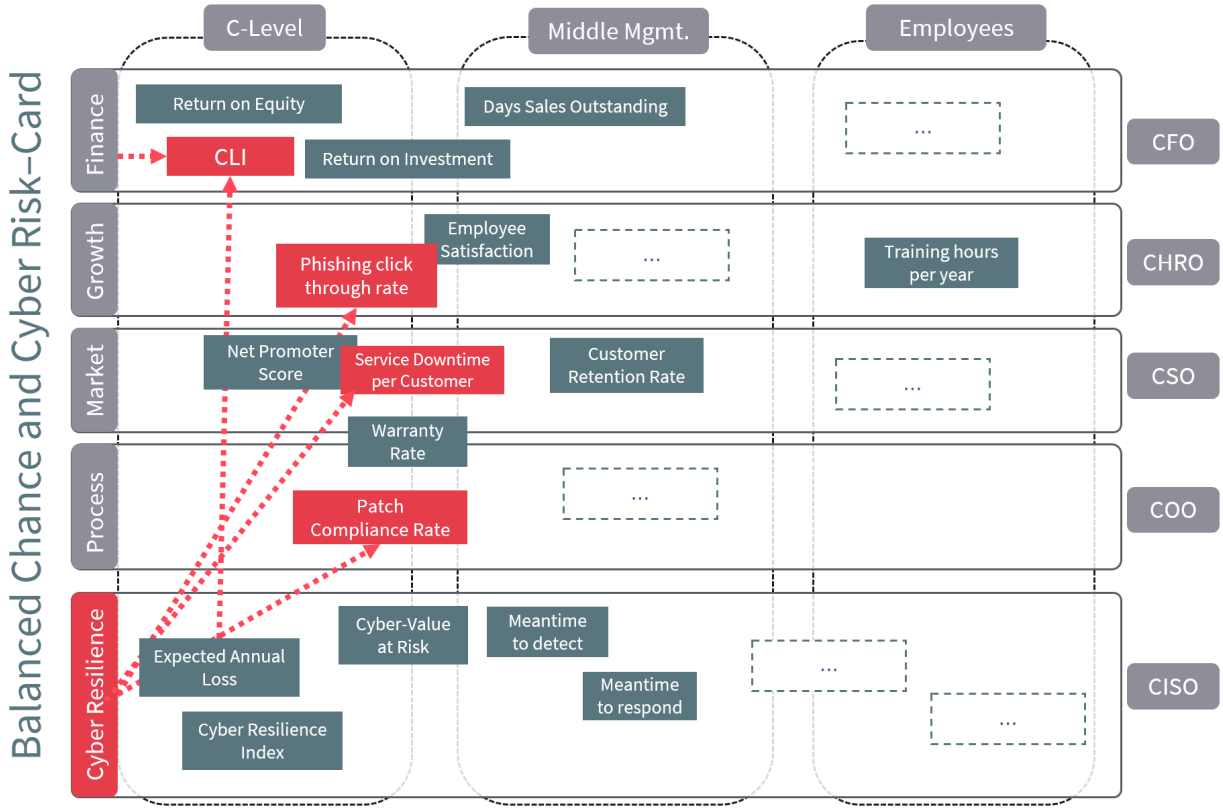


Figure 3. The Balanced Chance and Cyber-Risk Card with Integrated KPI/KRI Steering

assignment: the columns "C-Level," "Middle Management," and "Employees" indicate a cascading logic of indicators across organizational levels, while the executive roles on the right (CFO, CHRO, CSO, COO, CISO) suggest ownership of perspectives and indicator sets.

As a closing remark, the specific selection and operational definition of KPIs and KRIs must remain industry-dependent and should therefore be derived along the organization's value chain and key value-creation drivers to ensure decision relevance and comparability within the given business model.

All KPIs/KRIs should use a three-step traffic-light logic. The limits depend on the branch and the individual business, but as a rule of thumb, one can firstly go with the following suggestion:

- **Green:** on or better than target;
- **Yellow:** target–10% (warning);
- **Red:** $\geq$ 10% deviation, triggering an escalation to the next management tier and a liquidity stress-test in line with StaRUG early-warning duties.

E. Illustrative Cause-Effect Chains

To illustrate how cyber risk drivers and resilience capabilities propagate into business-relevant outcomes, this section presents four exemplary cause–effect chains based on the indicators introduced above. The chains are not intended as exhaustive models. They demonstrate how KRIs in the Cyber Resilience and Process perspectives can be translated into performance-relevant effects within the Market, Growth, and Finance perspectives, thereby supporting integrated steering and targeted management interventions.

1) Chain 1: Patching discipline to financial performance risk (CLI): A declining *Patch Compliance Rate* (Process KRI) widens the exposure window for known vulnerabilities, thereby increasing the likelihood of successful exploitation. This, in turn, elevates *EAL* and *CyVaR* (Cyber Resilience indicators) as the expected frequency and severity of incidents increase. As EAL rises relative to operating performance, *CLI* (Finance) increases, signaling that a larger share of operating earnings is at risk and that cyber exposure has become materially performance-relevant for executive steering. In regulated sectors, elevated cyber exposure may additionally increase the probability of non-compliance with regulatory obligations such as NIS2, sector-specific resilience requirements, or contractual security obligations, potentially resulting in financial penalties, liability claims, audit findings, or reputational damage. Consequently, insufficient patching discipline may create both operational cyber risk and regulatory governance risk. In an integrated steering cycle, this chain motivates prioritized remediation programs and governance escalation when patch compliance falls below defined thresholds.

2) Chain 2: Phishing susceptibility to operational disruption and customer impact: An increasing *Phishing Click-Through Rate* (Growth KRI) indicates rising susceptibility to social engineering, increasing the likelihood of credential

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

compromise and malware infiltration. This leads to a higher expected incident rate and increases the burden on detection and response functions, often reflected in deteriorating *MTTD* and *MTTR* (Cyber Resilience capability indicators). Longer detection and response cycles typically amplify incident impact and can translate into increased *Service Downtime per Customer* (Market impact indicator), negatively affecting *Net Promoter Score* and potentially reducing the *Customer Retention Rate* (Market KPIs). The chain thus explains how a human-factor risk driver can cascade into customer-facing outcomes and strategic market performance.

3) Chain 3: Resilience capability to market stability and retention: When *MTTD* and *MTTR* improve (Cyber Resilience), incidents are contained faster, and operational disruption is reduced. This resilience effect reduces *Service Downtime per Customer* (Market) and stabilizes the service experience, thereby supporting *Customer Retention Rate* and *Net Promoter Score* (Market KPIs). In parallel, faster containment reduces expected loss severity, lowering *EAL* and thereby improving *CLI* (Finance). This chain highlights that resilience investments can be justified not only through reduced risk exposure but also through measurable market-facing performance stabilization.

4) Chain 4: Capability building and culture to risk reduction and financial normalization: Higher *Training Hours per Year* (Growth KPI), combined with improved *Employee Satisfaction* (Growth KPI), can strengthen security-relevant behavior and compliance, thereby reducing the *Phishing Click-Through Rate* (Growth KRI) over time. Reduced susceptibility lowers the likelihood of compromise-driven incidents, supporting improved *MTTD* and *MTTR* (Cyber Resilience) and ultimately lowering *EAL*. As cyber exposure decreases relative to operating earnings, *CLI* (Finance) declines, signaling that operational performance is less threatened by cyber incidents. This chain illustrates how non-technical, people-oriented measures can materially influence cyber-risk exposure and produce economically interpretable effects in integrated steering.

Collectively, these illustrative cause-and-effect chains demonstrate how the proposed BCCR-Card enables a structured translation of cyber-risk signals into business-relevant steering pathways, linking risk drivers and resilience capabilities to market outcomes and financial performance exposure.

IV. Discussion

The BCCR-Card integrates cyber exposure into Reichmann's profitability-liquidity logic and extends concept-oriented Enterprise Performance Management toward cybersecurity-aware executive steering. Nevertheless, several limitations and research challenges must be acknowledged.

First, the framework relies on modeled frequencies and loss severities to quantify cyber risk exposure. Data scarcity, reporting bias, and limited loss transparency continue to constrain the statistical confidence of EAL and CyVaR estimates, particularly for low-frequency, high-impact events [44]. As a result, quantified outputs should be interpreted as decision-support approximations rather than precise forecasts.

Second, the traffic-light logic used for KPI and KRI escalation necessarily simplifies highly dynamic and adversarial cyber threat environments into discrete states. While this abstraction supports managerial usability and governance alignment, abrupt threshold effects may obscure early trend deterioration or nonlinear risk accumulation in fast-evolving attack scenarios.

Third, transferring the BCCR-Card across industries and organizational contexts requires careful recalibration of KPIs and KRIs. Indicator relevance, measurement cadence, and acceptable thresholds vary substantially between environments such as cloud-native organizations and Operational Technology-dominated settings, which limits direct cross-company benchmarking and necessitates context-sensitive customization.

Finally, the framework assumes a minimum level of organizational maturity in both controlling and cybersecurity functions. In organizations with fragmented data landscapes or unclear ownership between finance and security roles, implementation effort and coordination overhead may initially outweigh short-term steering benefits. These limitations highlight the importance of phased adoption and realistic expectation management when applying the BCCR-Card in practice. The proposed framework also differs conceptually from traditional Return on Security Investment (RoSI) approaches. Classical RoSI models primarily evaluate isolated security investments through static cost-benefit calculations and often struggle with uncertain probabilities, intangible losses, and rapidly changing threat dynamics [53], [54]. In contrast, the BCCR-Card is designed as a continuous enterprise steering and governance framework rather than a one-time investment valuation model. Its primary objective is not the precise financial valuation of individual security controls, but the integration of cyber-risk exposure, resilience capability, and organizational performance into recurring executive management processes through KPI/KRI coupling and risk-adjusted steering logic. Consequently, the framework emphasizes governance integration, escalation capability, and resilience-oriented management over deterministic investment-return calculations.

V. Adoption, Generalizability, and Lessons Learned

This section offers a concise methodology for SMEs, discusses generalizability beyond the German context, and summarizes lessons learned alongside future technical work from applying the BCCR-Card.

A. Methodology for Adoption

Effective use of the BCCR-Card starts with clear ownership and cadence at C-level, typically shared between finance and security leadership (e.g., CFO and CISO), with a monthly review focused on decisions rather than dashboards. Organizations select a small set of indicators. Ideally, no more than two per perspective to preserve attention on what moves value and resilience. Each indicator is defined in one sentence (scope, unit, and aggregation), assigned a single quarterly target, and governed by stable green/yellow/red thresholds to allow trend

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

interpretation. A minimal measurement layer reuses existing sources such as ticketing, endpoint management, SIEM, and backup reports, complemented by plausibility checks and temporary proxies where data coverage is incomplete. Two tabletop scenarios (e.g., ransomware and supplier outage) are used to translate operational signals into financial exposure via EAL and CyVaR. Approved actions are tracked against loss reduction and expenses, with quarterly reporting of EAL/EBIT and the ROCE of controls. In practice, SMEs can achieve a viable first loop within 90 days by fixing ownership and thresholds, assembling a single-page card with the most accessible data, running two scenarios, and replacing estimates with measured values as coverage improves.

For example, in a medium-sized manufacturing company, the BCCR-Card could integrate vulnerability-management telemetry from endpoint-management systems with ERP-derived operational performance indicators. A declining Patch Compliance Rate would increase the estimated EAL for ransomware scenarios and elevate CLI within the Finance perspective. This escalation could trigger predefined governance actions such as accelerated remediation programs, temporary investment reprioritization, or additional resilience controls. By embedding these signals into existing management review cycles, the framework allows cybersecurity exposure to become part of routine executive steering rather than a separate technical reporting activity.

B. Generalizability

While our examples reference German regulation, the design itself widely adopted frameworks (NIST CSF and ISO/IEC 27001) and governance principles compatible with COSO ERM. Sector characteristics primarily affect indicator choice (e.g., they choose the KPI liquidity ratio instead of ROCE) and data availability. The cause–and–effect logic and the financial coupling through EAL and CyVaR remain invariant. Very small companies can reduce scope to a single value stream without undermining the control logic.

C. Lessons Learned and Challenges

Across pilots, too many indicators weaken focus, frequent threshold changes falsify trends, and incomplete telemetry invites false precision if point estimates are reported without ranges. Persistently red signals require explicit decision mapping, a quick fix, structural control, or risk acceptance, to avoid loss. Finally, finance and security communities use different vocabularies. The BCCR-Card works as a shared language when explanations stay close to economics and risk awareness.

VI. Conclusion and Future Work

This paper introduced the Balanced Chance and Cyber-Risk Card as a concept-oriented steering framework for integrating cybersecurity into Enterprise Performance Management. Building on Reichmann's multidimensional controlling framework, Balanced Scorecard logic, and established corporate risk management principles, the BCCR-Card addresses a persistent gap in current practice: the weak linkage between technical cybersecurity metrics and executive-level performance steering.

The core contribution of this work lies in the systematic integration of cyber-specific Key Risk Indicators with conventional Key Performance Indicators within a unified, governance-ready steering architecture. By embedding cybersecurity as an explicit management perspective and coupling cyber-risk drivers, resilience capabilities, and economic impact measures with enterprise performance objectives, the framework enables a consistent translation of technical security signals into board-level, economically interpretable decision support. The introduction of Cyber Loss Intensity as a cross-perspective top indicator further strengthens this linkage by normalizing quantified cyber-loss exposure relative to operating performance, thereby making cyber risk directly comparable, prioritizable, and steerable alongside traditional financial metrics.

Beyond conceptual integration, the BCCR-Card provides a practical governance logic for executive decision-making. Cause–and–effect chains, scenario-based quantification, and role-specific indicator ownership support transparency, accountability, and escalation in line with regulatory requirements such as ISO 31000, COSO ERM, and emerging early-warning obligations. As such, the framework positions cybersecurity not as an isolated technical discipline, but as a measurable and manageable component of enterprise-wide value creation and risk control.

Future work should build on the conceptual contribution of this paper and address the identified limitations through four main directions. First, empirical validation through longitudinal multi-case studies is required to assess whether the proposed KPI/KRI signals reliably precede financial deviations and support superior risk-adjusted decision-making over time. Second, automation and data integration deserve further investigation, particularly the impact of integrating operational cybersecurity telemetry (e.g., SIEM, vulnerability management, backup systems) with financial and controlling systems on data quality, latency, and steering effectiveness. Third, advanced scenario modeling, including AI-supported threat and loss scenario generation, offers potential to improve tail-risk estimation and stress-testing under conditions of limited historical data. Finally, the growing convergence of cybersecurity with ESG and corporate governance reporting suggests extending the BCCR-Card toward integrated reporting contexts, enabling organizations to align cyber resilience with broader sustainability and governance objectives.

In summary, the BCCR-Card establishes a structured foundation for executive cyber risk governance by embedding cybersecurity into concept-oriented EPM. It offers both a rigorous analytical framework and a pragmatic steering instrument that bridges technical cyber risk management and economic performance control, paving the way for more resilient, transparent, and risk-aware enterprise leadership.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

REFERENCES

[1] A. Lawall and M. Drozdzynski, "The Balanced Chance & Cyber-Risk Card: Extending Reichmann's Multidimensional Controlling Framework for C-Level Steering in SMEs," in *SECURWARE 2025: The Nineteenth International Conference on Emerging Security Information, Systems and Technologies*. Barcelona, Spain: IARIA, Oct. 2025, pp. 76–82. [Online]. Available: https://www.thinkmind.org/articles/securware_2025_1_120_30046.pdf

[2] Bundesamt für Sicherheit in der Informationstechnik (BSI), "Die lage der it-sicherheit in deutschland 2025," Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn, Germany, Tech. Rep., Oct. 2025, accessed: 2026-01-17. [Online]. Available: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Cyber-Sicherheitslage/Lageberichte/Lagebericht/lagebericht_node.html

[3] Federal Bureau of Investigation (FBI) and Internet Crime Complaint Center (IC3), "2024 IC3 annual report," Federal Bureau of Investigation, Tech. Rep., 2025, accessed: 2026-01-17. [Online]. Available: https://www.ic3.gov/AnnualReport/Reports/2024_IC3Report.pdf

[4] UK National Cyber Security Centre (NCSC), "NCSC annual review 2025: It's time to act," UK National Cyber Security Centre, Tech. Rep., 2025, accessed: 2026-01-17. [Online]. Available: https://www.ncsc.gov.uk/files/ncsc-annual-review-2025.pdf

[5] European Union Agency for Cybersecurity (ENISA), "Enisa threat landscape 2025," European Union Agency for Cybersecurity (ENISA), Tech. Rep., 2025, accessed: 2026-01-17. [Online]. Available: https://www.enisa.europa.eu/sites/default/files/2025-10/ENISA%20Threat%20Landscape%202025.pdf

[6] National Cyber Security Centre (NCSC), Switzerland, "Semi-annual report 2025/1," National Cyber Security Centre (Switzerland), Tech. Rep., Nov. 2025, accessed: 2026-01-17. [Online]. Available: https://www.ncsc.admin.ch/dam/ncsc/de/dokumente/dokumentation/lageberichte/BACS-HJB-2025-1_EN.pdf.download.pdf/BACS-HJB-2025-1_EN.pdf

[7] Australian Signals Directorate (ASD) and Australian Cyber Security Centre (ACSC), "Annual cyber threat report 2024–2025," Web publication, Oct. 2025, accessed: 2026-01-17. [Online]. Available: https://www.cyber.gov.au/about-us/view-all-content/reports-and-statistics/annual-cyber-threat-report-2024-2025

[8] Cyber Security Agency of Singapore (CSA), "Singapore cyber landscape 2024/2025," Cyber Security Agency of Singapore, Tech. Rep., Sep. 2025, accessed: 2026-01-17. [Online]. Available: https://isomer-user-content.by.gov.sg/36/995dbbd7-a1de-4edb-b731-8fa36eb5546e/Singapore%20Cyber%20Landscape%202024_2025.pdf

[9] Bundesamt für Sicherheit in der Informationstechnik (BSI), "Lage-cybernation: Bedrohungen (monatslage)," Web publication, Nov. 2025, accessed: 2026-01-17. [Online]. Available: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Cyber-Sicherheitslage/Lageberichte/Monatsbericht_Lage-Cybernation/Lage_Bedrohungen/lage-Bedrohungen.html

[10] Horváth & Partners Management Consultants, "CFO Study 2024: The CFO's Path to a data Driven Company," Horváth & Partners, Stuttgart, Tech. Rep., Jul. 2024, [retrieved: July, 2025]]. [Online]. Available: https://www.horvath-partners.com/en/media-center/studies/cfo-study-2024-the-cfos-path-to-a-data-driven-company

[11] T. Reichmann and S. Form, "Balanced Chance- and Risk-Management," *Controlling – Zeitschrift für erfolgsorientierte Unternehmenssteuerung*, vol. 12, no. 4/5, pp. 189–198, 2000.

[12] T. Reichmann, "Die Balanced Chance- and Risk-Card: Eine Erweiterung der Balanced Scorecard [The Balanced Chance and Risk Card: An Extension of the Balanced Scorecard]," in *Risikomanagement nach dem KonTraG – Aufgaben und Chancen aus betriebswirtschaftlicher und juristischer Sicht*, K. W. Lange and F. Wall, Eds. München: Franz Vahlen, 2001, pp. 282–303.

[13] M. Diederichs, *Risikomanagement und Risikocontrolling [Risk Management and Risk Controlling]*, 5th ed. München: Franz Vahlen, 2023.

[14] European Parliament and Council of the European Union, "DIRECTIVE (EU) 2022/2555 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive)," 12 2022, [retrieved: May, 2026]]. [Online]. Available: https://eur-lex.europa.eu/eli/dir/2022/2555/oj

[15] ——, "REGULATION (EU) 2024/2847 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) No 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act)," 11 2024, [retrieved: May, 2026]]. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2024/2847/oj/eng

[16] G. Cokins, "The top seven trends in management accounting," *Edpacs*, vol. 53, no. 4, pp. 1–7, 2016.

[17] P. Horváth, R. Gleich, and M. Seiter, *Controlling*. Vahlen, 2015.

[18] T. Reichmann, M. Kißler, and U. Baumöl, *Controlling mit Kennzahlen: Die systemgestützte Controlling-Konzeption [Controlling with Key Figures: The System-Supported Controlling Concept]*, 9th ed. München: Franz Vahlen, 2017.

[19] J. Weber and U. Schäffer, *Einführung in das Controlling*. Schäffer-Poeschel, 2022.

[20] M. Drozdzynski, *Das neue Business Intelligence-gestuetze Krankenhaus-Controlling [The New Business Intelligence-Supported Hospital Controlling]*. Baden-Baden: Nomos, 2020.

[21] M. Liebe and M. Drozdzynski, "IT-gestützte Ausgestaltung einer GeSo-spezifischen mehrdimensionalen Controlling-Konzeption [IT-Supported Design of a Health and Social Care-Specific Multidimensional Controlling Concept]," *Controlling – Zeitschrift für erfolgsorientierte Unternehmenssteuerung*, vol. 30, no. 4, pp. 31–40, 2018.

[22] R. S. Kaplan and D. P. Norton, "The Balanced Scorecard – Measures That Drive Performance," *Harvard Business Review*, vol. 70, no. 1, pp. 71–79, 1992.

[23] R. S. Kaplan, D. P. Norton *et al.*, "Using the balanced scorecard as a strategic management system," 1996.

[24] R. S. Kaplan, "Conceptual foundations of the balanced scorecard," Tech. Rep., 2009.

[25] S. Kumar, W. M. Lim, R. Sureka, C. J. C. Jabbour, and U. Bamel, "Balanced scorecard: trends, developments, and future directions," *Review of managerial science*, vol. 18, no. 8, pp. 2397–2439, 2024.

[26] A. Silva, I. Maldonado, M. da Silva, and C. Cepeda, "Sustainability balanced scorecard: Systematic literature review." *Journal of Risk & Financial Management*, vol. 18, no. 6, 2025.

[27] T. C. Herath, H. S. Herath, and D. Cullum, "An information security performance measurement tool for senior managers: Balanced scorecard integration for security governance and control frameworks," *Information Systems Frontiers*, vol. 25, no. 2, pp. 681–721, 2023.

[28] P. Hopkin and C. Thompson, *Fundamentals of Risk Management: Understanding, Evaluating and Implementing Effective Enterprise Risk Management*, 6th ed. London: Kogan Page, 2022.

[29] *Risk Management-Guidelines (ISO 31000:2018)*, International Organization for Standardization Std., Feb. 2018.

[30] M. Diederichs, "Balanced Chance- & Risk-Card," *Controlling*, vol. 16, no. 12, pp. 703–705, 2004.

[31] ISACA. (2024, Apr.) How CISOs Can Take Advantage of the Balanced Scorecard Method. [retrieved: July, 2025]. [Online]. Available: https://www.isaca.org/resources/news-and-trends/industry-news/2024/how-cisos-can-take-advantage-of-the-balanced-scorecard-method

[32] M. Van De Ven, P. Lara Machado, A. Athanasopoulou, B. Aysolmaz, and O. Turetken, "Key performance indicators for business models: a systematic review and catalog," *Information Systems and e-Business Management*, vol. 21, no. 3, pp. 753–794, 2023.

[33] D. Parmenter, *Key performance indicators: developing, implementing, and using winning KPIs*. John Wiley & Sons, 2015.

[34] Institute of Operational Risk (IOR), "Key risk indicators: Sound practice guidance," Institute of Operational Risk, Tech. Rep., Nov. 2010, accessed: 2026-02-04. [Online]. Available: https://www.ior-institute.org/public/IORKRIGuidanceNov2010.pdf

[35] G. J. van den Brink and M. Leipoldt, "Key risk indicators reloaded," *Maandblad voor Accountancy en Bedrijfseconomie*, vol. 96, no. 5/6, pp. 165–171, 2022, accessed: 2026-02-04. [Online]. Available: https://mab-online.nl/article/80730/

[36] A. Lawall and P. Beenken, "A Threat-Led Approach to Mitigating Ransomware Attacks: Insights from a Comprehensive Analysis of the Ransomware Ecosystem," in *Proceedings of the 2024 European Interdisciplinary Cybersecurity Conference*, ser. EICC '24, S. Li, K. Coopamootoo, and M. Sirivianos, Eds. New York, NY, USA: Association for Computing Machinery, 2024, pp. 210–216. [Online]. Available: https://doi.org/10.1145/3655693.3661321

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[37] R. Riesco and V. A. Villagrá, "Leveraging cyber threat intelligence for a dynamic risk framework: Automation by using a semantic reasoner and a new combination of standards (stix™, swrl and owl)," *International Journal of Information Security*, vol. 18, no. 6, pp. 715–739, 2019.

[38] C. Sánchez-Zas, V. A. Villagrá, M. Vega-Barbas, X. Larriva-Novo, J. I. Moreno, and J. Berrocal, "Ontology-based approach to real-time risk management and cyber-situational awareness," *Future Generation Computer Systems*, vol. 141, pp. 462–472, 2023.

[39] R. Graf, F. Skopik, and K. Whitebloom, "A decision support model for situational awareness in national cyber operations centers," in *2016 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*. IEEE, 2016, pp. 1–6.

[40] T. Väisänen, S. Noponen, O.-M. Latvala, and J. Kuusijärvi, "Combining real-time risk visualization and anomaly detection," in *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, 2018, pp. 1–7.

[41] J. Smallman, "The effectiveness of cyber threat intelligence in improving security operations," *Journal of Artificial Intelligence General science (JAIGS) ISSN:3006-4023*, vol. 5, no. 1, p. 189–209, Jul. 2024. [Online]. Available: https://ojs.boulibrary.com/index.php/JAIGS/article/view/193

[42] H. Kure and S. Islam, "Cyber threat intelligence for improving cybersecurity and risk management in critical infrastructure," *Journal of Universal Computer Science*, vol. 25, no. 11, pp. 1478–1502, 2019.

[43] R. A. Firdaus, N. A. Rakhmawati, and F. Samopa, "A state-of-the-art review of cyber threat intelligence awareness programs in mitigating bank cyber attacks," in *2024 IEEE International Symposium on Consumer Technology (ISCT)*. IEEE, 2024, pp. 648–654.

[44] M. Eling and W. Schnell, "Ten Key Questions for Cyber Risk Modelling: A Systematic Review and Research Agenda," *The Geneva Papers on Risk and Insurance-Issues and Practice*, vol. 47, no. 3, pp. 366–401, 2022.

[45] S. Myerson, "Why Heat Maps Fail for Cybersecurity Risk Management," Gartner, Inc., Research Note G00734212, September 2023.

[46] European Union Agency for Cybersecurity (ENISA), "ENISA Threat Landscape 2024," Athens, Tech. Rep., 2024, [retrieved: July, 2025]. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024

[47] M. Eling, U. Kühn, and H. Gruber, "Modelling Systemic Cyber Risk: Accumulation and Cascade Effects in Interconnected IT Ecosystems," *Journal of Risk Finance*, vol. 24, no. 1, pp. 1–26, 2023.

[48] R. N. Anthony and V. Govindarajan, *Management Control Systems*, 13th ed. New York: McGraw-Hill Education, 2017.

[49] R. Simons, *Levers of Control: How Managers Use Innovative Control Systems to Drive Strategic Renewal*. Boston: Harvard Business School Press, 1995.

[50] K. A. Merchant and W. A. V. der Stede, *Management Control Systems: Performance Measurement, Evaluation and Incentives*, 5th ed. Harlow: Pearson, 2021.

[51] Bundesrepublik Deutschland, "Aktiengesetz (AktG) [Stock Corporation Act]," Bundesgesetzblatt I 1965, S. 1089; zuletzt geändert durch Art. 1 Gesetz vom 21. Dez. 2023, BGBl. I 2023 Nr. 394, Sep. 1965, [retrieved: July, 2025]. [Online]. Available: https://www.gesetze-im-internet.de/aktg/BJNR010890965.html

[52] ——, "Gesetz über den Stabilisierungs- und Restrukturierungsrahmen für Unternehmen (Unternehmensstabilisierungs- und -restrukturierungsgesetz – StaRUG) [Law on the Stabilization and Restructuring Framework for Companies (Corporate Stabilization and Restructuring Act – StaRUG)]," Bundesgesetzblatt I, Nr. 66, 29. Dez. 2020, S. 3256–3340, Dec. 2020, [retrieved: July, 2025]. [Online]. Available: https://www.gesetze-im-internet.de/starug/BJNR325610020.html

[53] A. Davis, "Return on Investment: Return on security investment - proving it's worth it," *Network Security*, vol. 2005, no. 11, p. 8–10, Nov. 2005. [Online]. Available: https://doi.org/10.1016/S1353-4858(05)70301-9

[54] T. Yaqoob, A. Arshad, H. Abbas, M. F. Amjad, and N. Shafqat, "Framework for Calculating Return on Security Investment (ROSI) for Security-Oriented Organizations," *Future Generation Computer Systems*, vol. 95, no. C, p. 754–763, Jun. 2019. [Online]. Available: https://doi.org/10.1016/j.future.2018.12.033

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Practical Evaluation of Password-Authenticated Key Exchange Schemes in Classical and Post-Quantum Settings on Android

Jörn-Marc Schmidt
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
email: joern-marc.schmidt@iu.org

Alexander Lawall
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
email: alexander.lawall@iu.org

***Abstract*—Password-Authenticated Key Exchange (PAKE) protocols enable secure key establishment using low-entropy passwords and are widely deployed in mobile and networked systems. Most deployed PAKEs rely on classical cryptographic assumptions that are vulnerable to quantum adversaries, raising the need for post-quantum alternatives whose practical feasibility on mobile platforms remains unclear. This paper investigates whether post-quantum PAKE schemes can be efficiently deployed on modern Android devices and how they compare to classical PAKE protocols. We implement a representative set of balanced and augmented PAKE schemes, including classical and post-quantum constructions based on the One-Way Key Encapsulation Method to PAKE (OCAKE), using Kotlin/Java and the Bouncy Castle cryptographic provider. All schemes are evaluated on a physical Google Pixel 7 Pro device under controlled conditions using the Android Jetpack Microbenchmark framework. We measure execution time, message sizes, and energy consumption for security levels corresponding to 128-bit and 256-bit classical security. The results show that post-quantum PAKEs achieve competitive computational performance on mobile devices, at the cost of increased communication overhead, demonstrating their practical viability for post-quantum-ready authentication systems.**

Keywords-PAKE; post-quantum cryptography; Android; password-based authentication; mobile security

I. Introduction

This work is an extended version of *Comparison of Password-Authenticated Key Exchange Schemes on Android*, published at SECURWARE 2025 [1]. Typical credentials for authentication are passwords. They can be remembered and typed in by humans on various input devices. In terms of security, however, they are commonly easier to guess or to brute-force than cryptographic keys. Password-Authenticated Key Exchange (PAKE) schemes address this issue. They are interactive protocols for two or more parties to generate a joint session key based on a shared password. An adversary eavesdropping on the connection cannot discover the password. Active attacks are, in the best case, limited to one possible password guess per protocol run.

Hence, PAKE schemes can improve security in various cases of password use. For example, a PAKE is employed for password-based authentication in Wireless Fidelity (Wi-Fi) networks [2] and the Matter protocol [3]. Different applications, including 1Password [4] and messengers, such as WhatsApp [5] and Facebook Messenger [6] make use of such schemes. Furthermore, Apple relies on PAKE protocols for HomeKit device enrollment [7], iCloud Keychain escrow [8], and in the Car Key pairing process [9].

However, the PAKE schemes that are used in practical applications rely on the discrete logarithm problem, its elliptic-curve variant, or similar problems. As these problems cannot be considered secure in the presence of cryptographically relevant quantum computers, new protocols are required in this regard. Potential solutions are generic ways to transfer primitives, such as Key Encapsulation Mechanisms (KEMs) into secure PAKE protocols [10]. The resulting OCAKE scheme was recently implemented by Alnahawi et al. on SmartMX3 P71D600 smart card [11]. Lyu et al. published a method for transforming balanced PAKE schemes, in which both parties know the password, into asymmetric or augmented PAKE schemes suitable for client-server settings [12]. The authors also applied their method to lattice-based schemes, yielding lattice-based post-quantum-secure protocols.

In general, the design and analysis of PAKE schemes is an ongoing effort. The work of Alnahawi et al. lists 30 balanced schemes and 19 augmented schemes that have been published since 2015 [13]. They come with different security proofs and various levels of analysis by other researchers. Several also provide benchmark figures for their specific implementations.

In this paper, PAKEs with known real-world applications are implemented for Android devices and their performance is measured and compared. In particular, we selected Dragonfly as a balanced PAKE scheme for its use in Wi-Fi Protected Access 3 (WPA3), Password-Authenticated Connection Establishment (PACE) as implemented in travel documents, and CPACE as it is used by Facebook Messenger. We implemented the One-Way Key Encapsulation Method to PAKE (OCAKE) scheme, using Module-Lattice Key Encapsulation Mechanism (ML-KEM) to also include post-quantum-secure schemes. For comparison with classical constructions, One-Encryption Key Exchange (OEKE) was implemented as well.

For augmented schemes, the Secure Remote Password (SRP) protocol was chosen due to the possibility of integration with Transport Layer Security (TLS) [14] and its use with 1Password. In addition, SPAKE2+, together with its balanced version SPAKE2, was selected as it is used by Apple HomeKit, Apple Car Key, and the Matter protocol. We also applied the transformation of Lyu et al. to OCAKE, to evaluate a post-quantum-secure augmented PAKE scheme.

The remainder of the paper is organized as follows. Section II gives a brief overview of the implemented PAKE schemes and their properties. The details of the test setup are given in

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

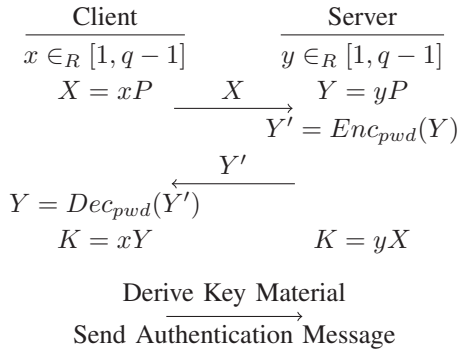


Figure 1. Simplified version of OEKE [18], using a shared password *pwd* and a generator P of order q of an additive group

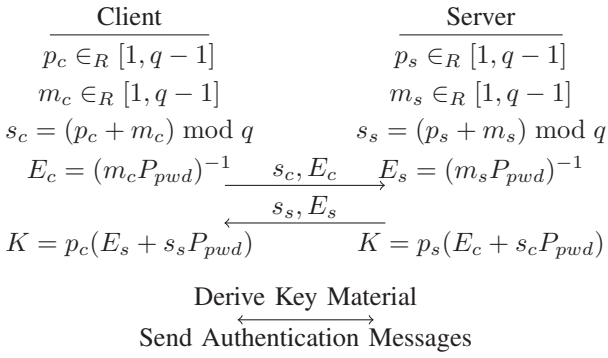


Figure 2. Simplified version of Dragonfly [19], using a shared password element P_{pwd} in an additive group of order q

Section III, before the results, together with implementation-specific choices, are presented in Section IV. Conclusions are drawn in Section V.

II. PAKE Schemes

Details on how a PAKE achieves its objective of agreeing on a strong cryptographic key based on a shared password differ from scheme to scheme. In general, it is possible to distinguish between balanced schemes, where both parties know the password, and augmented schemes, where one party, often called the prover, knows the password and the other party, often called the verifier, possesses a verification value but not the password itself. This prevents the verifier from impersonating the prover towards another third party. A discussion of the properties of PAKE schemes and related security considerations can be found in [15].

The following notation will be used in the remainder of the paper. Let $Enc_{key}(\cdot)$ and $Dec_{key}(\cdot)$ denote encryption and decryption of a message with a symmetric cipher using a shared key. A hash function is denoted as $H(\cdot)$. The simplified protocols show only the agreement of a shared secret – in order to derive key material, further steps, like applying key derivation functions, are required. In addition, every protocol requires a verification phase to ensure that both parties followed the protocol and agreed on the same key material. This can be achieved by exchanging hash or message authentication code (MAC) values over the protocol messages.

In 1992, Bellovin and Merritt published the first PAKE called *Encrypted Key Exchange (EKE)* [16]. It is a balanced scheme. Following different security analyses, including [17], the variant *One-Encryption Key Exchange (OEKE)* was proposed [18]. The underlying idea of these schemes is to derive a secret key from the password and use it to encrypt a public key that is used for key agreement. Hence, the receiver requires the password to decrypt the public key and continue with the protocol. Figure 1 gives a simplified version of the scheme. EKE and OEKE are the inspiration for various post-quantum (PQ) PAKE schemes, including OCAKE [10]. Instead of encrypting a key for a (Elliptic Curve) Diffie-Hellman key agreement method with the password, it uses a KEM and encrypts the related public key with the password.

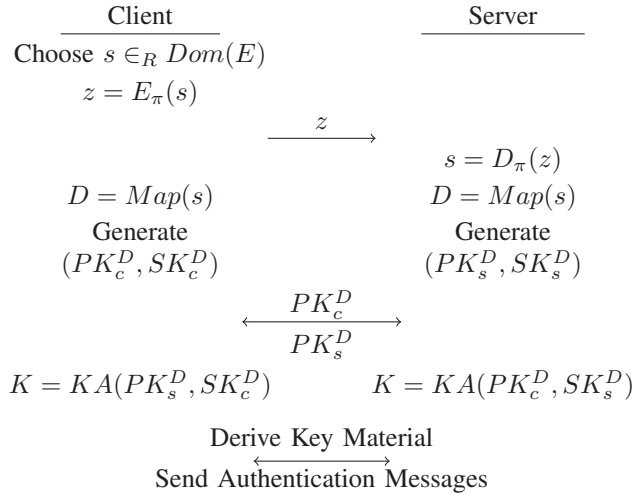


Figure 3. Simplified PACE Scheme.

Another approach is followed by the protocol Dragonfly, defined in RFC 7664 [19]. It maps the password to a group element and uses a random mask to blind it. This blinded value is exchanged and can be used for the next steps towards key agreement. The mapping into the group element requires a specific process. RFC 7664 defines an algorithm called *Hunting and Pecking*, which searches for such an element. In order to ensure a time-constant behavior to prevent side-channel leakage, a constant number of attempts is made. In addition, critical checks are masked using random values. A Dragonfly sample run is shown in Figure 2.

Another PAKE that relies on mapping the password to a group element is the PACE protocol used in travel documents [20]. PACE relies on a random value that is encrypted using the shared password. After mapping the value to the group, a key agreement is performed. A protocol flow is shown in Figure 3. For mapping the point to the group, the standard mandates support of at least two mechanisms, a) the *Generic*

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Mapping based on an (Elliptic Curve (EC)) Diffie-Hellman key agreement, and b) the *Integrated Mapping*, which directly maps a value into the group. A third version, called *Chip Authentication Mapping* is optional.

A similar approach is followed by the Composable Password Authenticated Connection Establishment (CPACE) [21] scheme. Both parties share group parameters and a password. For the mapping, a function specified in RFC 9380 [22] should be used, which corresponds to the algorithm used for the Integrated Mapping. Its first step at both ends is to derive a group element from the password, instead of choosing a random value as input for the mapping as in the PACE protocol. The mapped element is then used for key agreement using a (EC) Diffie-Hellman protocol.

Client		Server
$x \in_R [1, q-1]$		$y \in_R [1, q-1]$
$X = xP$		$Y = yP$
$p_A = wM + X$	$\xrightarrow{p_A}$	$p_B = wN + Y$
	$\xleftarrow{p_B}$	
$K = x(p_B - wN)$		$K = y(p_A - wM)$

Derive Key Material

Send Authentication Messages

Figure 4. Simplified version of SPAKE2 [23], using a password element w, a generator P of order q of an additive group G and fixed elements M, N

SPAKE2 follows a different approach [23]. It does not require such a mapping function but allows one to create two elements of the used group beforehand. They are defined for the set of parameters used and are independent of the password. The password is mapped to a scalar using a memory-hard hash function (e.g., scrypt or PBKDF2). This scalar is blinded using random values and the predefined elements on both ends. The results are exchanged, allowing to agree on a shared element. Its flow is shown in Figure 4.

In applications with a clear client-server relationship, an asymmetric or augmented PAKE can be beneficial. Those schemes provide the server or verifier with the possibility to ensure that the client or prover knows the password, without being able to impersonate the client. This requires a registration step where a record that is stored by the verifier is generated.

An example of such an augmented PAKE is SRP. The registration phase of the SRP scheme consists of hashing the password together with a salt value and generating a password verifier in a prime field, using the hashed value as exponent of a group generator. The verifier knows only the verification value; the prover can generate it using the password and the salt. This allows the parties to perform a key agreement based on the password. The flow of the process is shown in Figure 5. Note that SRP relies on multiplying group elements and exponentiating elements in a finite field, which does not map straightforwardly onto elliptic curve groups. Hence, the protocol does not have a simple elliptic curve analogue.

Registration

$s \in_R [1, 2^{64}]$

$x = H(s|H(U|":"|p))$

$v = g^x$

Store(s, v)

Authentication

Client		Server
$a \in_R [1, q-1]$		$b \in_R [1, q-1]$
$A = g^a$	$\xrightarrow{A}$	$B = v + g^b$
	$\xleftarrow{s, B}$	
$u = H(A, B)$		$u = H(A, B)$
$x = H(s \parallel pwd)$		
$e = (a + u * x)$		
$K = (B - g^x)^e$		$K = (A * v^u)^b$

Derive Key Material

Send Authentication Messages

Figure 5. Simplified version of SRP [24], using a finite field G with a generator g of order q

SPAKE2+ [25] is an augmented version of SPAKE2. The authentication flow is similar to SPAKE2. The augmented part is achieved via a registration step, producing a registration record. This record is used in the protocol by the verifier that does not know the password itself.

In order to transform a balanced post-quantum PAKE into an augmented post-quantum PAKE, the transformation of Lyu et al. can be applied [12]. In addition to the balanced PAKE scheme, it uses a KEM and authenticated encryption. During the registration phase, the password is hashed and used to generate a key pair for the KEM scheme, while only the hash and public key are stored. During the authentication phase, the hashed password is used as input to the balanced PAKE to agree on a key. This is followed by an authentication process where the client derives the KEM key pair from the password by following the steps of the registration procedure. Hence, the client now has the private key to the public key contained in the registration record. This enables finalization of the protocol by ensuring that the client knows the password. Using a quantum-secure balanced PAKE and a post-quantum KEM, the protocol provides quantum security.

III. Testing Setup

All benchmarks were performed on a Google Pixel 7 Pro smartphone, that is, the physical device, not the emulator. This device features a Google Tensor G2 SoC, 12 GB RAM, and runs the Android 16 operating system. Its hardware and up-to-date system software ensure that performance measurements are representative of modern Android platforms. The Pixel 7 Pro device was connected to a Windows test PC via USB with

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

developer options enabled, airplane mode turned on, and all connectivity (Wi-Fi, Bluetooth, mobile data) disabled to reduce interference. The battery saver mode and adaptive battery features were kept off. Using the developer options, the limit for background processes was set to zero. The implementations are written in Kotlin/Java and use, in addition to native libraries, a Bouncy Castle provider in version 1.81. Their build target was Android API 34 and ProGuard/R8 minification was enabled. In addition to the pure Java implementations, versions that use optimized OpenSSL scalar multiplications for elliptic curve were tested as well. The OpenSSL-based scalar multiplication was around 3 times faster compared to the pure Java Bouncy Castle-based scalar multiplication for a 128-bit level. For 256-bit security level, i.e., for secp521r1, no performance gains from the OpenSSL version were measured.

For performance measurements, microbenchmarking using the Android Jetpack Benchmark Library (version 1.3.4) was used [26]. The library orchestrates test runs, performs warm-up iterations, and leverages the platform's trace-based timing mechanism to reliably capture execution durations. The benchmark process pins the test process to a foreground priority and requests sustained performance mode to reduce CPU/GPU thermal throttling. During this process, 50 measurement runs are conducted. The whole measurement was repeated 100 times, leading to 5000 data points per test. However, it turned out that the measurements contained some outliers, as it was not possible to prevent all side effects during the measurement process. In order to cope with those, measurements that took more than 10 times the mean of the current set are removed. After this procedure, every set contains between 4,974 and 5,000 data points, on average 4,993 data points. The following results are the mean values and the 95% confidence interval of these measurements.

Energy consumption is measured using test functions that cover the whole scheme, that is, client and server operations. For every microbenchmark of a test function, which involves 50 runs of that function, a Perfetto trace [27] is created and analyzed. In particular, the power rail data for the large central processing unit (CPU) core is used. The data points associated with a function run are extracted, yielding an accumulated energy consumption value. Hence, the delta between the first and last measurement run a data point is available, gives the number of measurements that are conducted using the energy given by the value difference of those data points. Note that not every power trace contains two values that can be associated with measurement runs. In such cases, the whole run, i.e., this microbenchmark of all PAKE schemes, is discarded. Otherwise, this procedure gives an energy consumption data point for every test function, each covering a complete PAKE scheme. In order to prevent thermal effects on the power consumption, the order of the tests within a microbenchmark is randomized. Overall, 250 measurements were conducted and evaluated for every scenario, i.e., for 128 and 256-bit security levels, each using pure Java implementations and with OpenSSL optimizations. In every measurement run, some traces were discarded due to missing data points, resulting in 222-247 energy consumption results per function to be evaluated.

IV. Implementation and Results

In order to allow a fair comparison of the different implementations, all underlying primitives were chosen with parameters providing a consistent security level. In particular, for 128-bit security, Advanced Encryption Standard (AES) with 128-bit keys for encryption, Secure Hash Algorithm (SHA) 2 with a 256-bit digest as hash function, a discrete logarithm group with 3072 bits [28], secp256r1 [29] as 256-bit elliptic curve and ML-KEM512 [30] were used. In order to show the differences in terms of performance for another security level, the same implementations were tested with parameters that provide 256-bit security. Here, a discrete logarithm group with 8192 bits [28] and secp521 [29] are used together with AES with a 256-bit key and SHA-2 with a 512 bit output and ML-KEM1024 [30] were selected.

For modular operations, the native BigInteger library is used. For cryptographic algorithms, native implementations like `javax.crypto.Cipher` for AES and `java.security.MessageDigest` for SHA are employed. For all others, including HMAC, HKDF, PBKDF2, scrypt, ML-KEM, and ECC operations, implementations provided by Bouncy Castle are used. In this context, Bouncy Castle uses a Window Non-Adjacent Form (WNAF) multiplier for EC scalar multiplication. The OpenSSL version uses optimized implementations for the specific curves. For the implementations, the client and server components were tested on the same device. All elements that require transfer between the parties were encoded as byte arrays; compression was used for elliptic curve points.

As described in Section II, several schemes, especially those based on Elliptic Curve Cryptography (ECC), require a mapping from a random string to a point in the group used. Hence, different mapping functions were implemented. In particular, *Hunting and Pecking* as specified in RFC 7664 [19], *Generic Mapping*, and *Integrated Mapping* as specified in [20].

TABLE I
Performance of different mapping functions from a random value to a point on secp256r1 / secp521r1.

Mapping	Time in μs	
	secp256r1	secp521r1
Hunting and Pecking	$4,352 \pm 4.1$	$9,585 \pm 7.1$
Generic Mapping	700 ± 0.5	$5,335 \pm 2.8$
Integrated Mapping	62 ± 0.0	253 ± 0.2

Note that potential overhead for the generic mapping for exchanging messages is not reflected in this number.

A performance comparison can be found in Table I. Note that the Generic Mapping requires a message exchange between two parties to agree on a common result of the mapping. The figures in the table only reflect the computational effort and not the potential network latency for exchanging those messages. The Hunting and Pecking was configured with a minimum of 40 iterations as suggested in the respective RFC. If the iteration ends as soon as a suitable element is found, the

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE II
OVERVIEW OF USED PARAMETERS FOR IMPLEMENTED SCHEMES

Scheme	Group	Cipher	Hash	Mapping	Others
Balanced Schemes					
OEKE	3072-bit/8192-bit MODP Group	AES-CBC (128/256)	SHA-2 (256/512)	-	-
OEKE (ECC)	secp256r1/secp521r1	AES-CBC (128/256)	SHA-2 (256/512)	-	-
OCAKE	ML-KEM (512/1024)	AES-CBC (128/256)	SHA-2 (256/512)	-	PBKDF2-HMAC-SHA-2
Dragonfly	secp256r1/secp521r1	-	SHA-2 (256/512)	Hunting and Pecking	HKDF-SHA-2
SPAKE2	secp256r1/secp521r1	-	SHA-2 (256/512)	-	HKDF-SHA-2
PACE (IM)	secp256r1/secp521r1	AES-CBC (128/256)	SHA-2 (256/512)	Integrated Mapping	PBKDF2-HMAC-SHA-2
PACE (GM)	secp256r1/secp521r1	AES-CBC (128/256)	SHA-2 (256/512)	Generic Mapping	PBKDF2-HMAC-SHA-2
CPACE	secp256r1/secp521r1	-	SHA-2 (256/512)	Integrated Mapping	-
Augmented Schemes					
SRP	3072-bit/8192-bit MODP Group	-	SHA-1/ SHA-2 (256/512)	-	-
SPAKE2+	secp256r1/secp521r1	-	SHA-2 (256/512)	-	HKDF-SHA-2/scrypt
aPAKE-PQC	ML-KEM (512/1024)	AES-GCM (128/256)	SHA-2 (256/512)	-	-

result is obtained on an average of $645 \pm 0.9\mu s$ for secp256r1 and $882 \pm 1.1\mu s$ for secp521r1; however, this approach does not ensure constant-time execution. As the Integrated Mapping computes the result directly–without iterative steps like Hunting and Pecking or key generation and exchange as in Generic Mapping–it is, as expected, the most efficient among the evaluated methods. The relation between the mapping schemes is the same for secp256r1 and secp521r1. However, while the runtime of the Hunting and Pecking method roughly doubles, the runtime of the Integrated Mapping increases by a factor of 4. The runtime of the Generic Mapping even increases by a factor larger than 7 when the security level is doubled.

In order to compare the performance of balanced PAKEs that provide quantum security and those relying on *traditional* asymmetric primitives, OEKE was implemented and tested using a discrete logarithm group and an elliptic curve. OCAKE was tested using ML-KEM.

The implementation of OEKE follows the definition in [18]. For a 128-bit security level, instances using a discrete logarithm group with 3072 bits as defined in RFC 3526 [28], and using secp256r1 [29] are evaluated. Both use AES-128 with Cipher Block Chaining (CBC) for the encryption of the public key with the password and SHA-2 with 256-bits digest for computing the authentication tags. The realization of OCAKE is based on the design of Beguinet et al. [10]. A secret key is derived from the shared password using Password-Based Key Derivation Function 2 (PBKDF2) with Hash-based Message Authentication Code (HMAC). This key is used to encrypt and transfer a ML-KEM512 public key. Again, SHA-2 is used to compute the authentication tags, ensuring that both parties derive the same key material. For a 256-bit security level, a discrete logarithm group with 8192 bits, secp521r1, AES-256 and SHA-2 with a 512 bit output was used.

Using elliptic curves provides a notable performance advantage compared to a discrete logarithm group, while the OCAKE implementation is even faster. As the client in the OCAKE protocol needs to generate a key pair, it requires more computational effort than the server. In detail, OEKE required $4,948 \pm 1.1\mu s$/$4,955 \pm 1.5\mu s$ on the client/server compared to $535 \pm 0.7\mu s$/$507 \pm 0.6\mu s$ when using an elliptic curve for 128-bit security. For a security level of 256-bits, the runtime goes down from $293,212 \pm 23.3\mu s$/$293,520 \pm 19.3\mu s$ to $3,714 \pm 4.1\mu s$/$3,566 \pm 4.0\mu s$. Hence, where there is a runtime increase of a factor of about 60 for the MODP based version, the ECC version of OEKE requires around 7 times more runtime when switching from a 128-bit to a 256-bit security level. When using the OpenSSL optimizations, the OEKE(ECC) client/server process required $343 \pm 0.3\mu s$/$315 \pm 0.3\mu s$ for 128-bit security level, while for 256-bit, no significant changes are observed ($3,741 \pm 4.1\mu s$/$3,636 \pm 3\mu s$ for client/server operations). The OCAKE scheme completed in $242 \pm 0.2\mu s$/$151 \pm 0.2\mu s$ for 128-bit security level and in $490 \pm 0.4\mu s$/$298 \pm 0.3\mu s$ for 256-bit security level, i.e., the runtime increases only by a factor of 2.

However, this speedup comes at the cost of larger messages, as OCAKE requires exchanging the encrypted KEM public key and the ciphertext of 16+816+768 bytes in addition to two 32-byte authentication tags. In contrast, OEKE(ECC)/OEKE exchanges one public key of 33/384 bytes, one encrypted public key of 16+48/16+400 bytes in addition to a 32-byte authentication tag. For the 256-bit version, OEKE(ECC)/OEKE exchange public keys of 67/1024 bytes and 16+80/16+1040 bytes in addition to a 64 byte authentication tag. OCAKE requires exchanging 16+1584+1568 bytes plus 2 authentication tags of 64 bytes each.

Dragonfly was implemented according to RFC [19] using secp256r1 for 128-bit security and secp521r1 for 256-bit security. It uses the Hunting and Pecking mapping with at least 40 iterations. For derivation of the secret from the negotiated point, HMAC-based Key Derivation Function (HKDF) SHA-2 with a 256/512 bit digest is used. In order to blind specific checks of the protocol, specific elements need to be found. This can be done once during an initialization phase that took $294 \pm 0.5\mu s$ for secp256r1 and $649 \pm 1.1\mu s$ for secp521r1.

As client and server perform the same operations, they require the same time. If blinding is omitted and a non-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

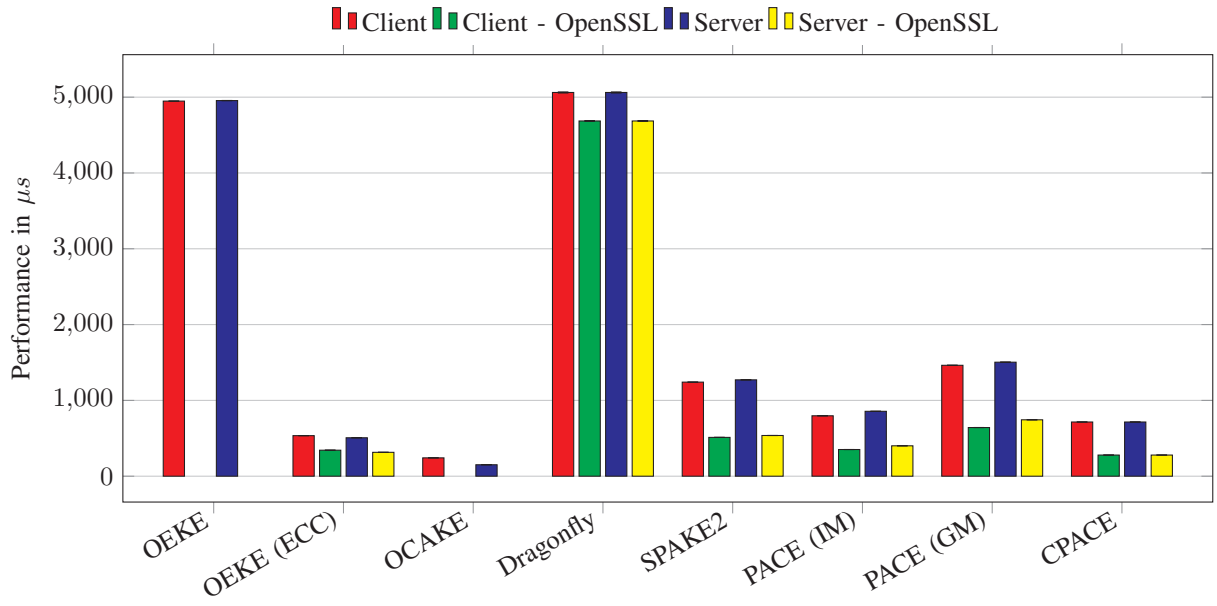


Figure 6. Performance Comparison of the balanced PAKE Schemes for 128-bit security level

constant-time mapping is used, the agreement process could be reduced from $5,062 \pm 6.8\mu s$ to $1,327 \pm 1.5\mu s$. for the 256-bit security version, the runtime with blinding is $16,381 \pm 22.3\mu s$, without $8,235 \pm 10.3\mu s$. In terms of message size, Dragonfly exchanges a message containing a scalar and an ECC point from server to client and the other way round, i.e., $2 \times (32+33)$ bytes / $2 \times (66 + 67)$ in addition to two authentication tags of 32/64 bytes each for 128/256-bit security level. The OpenSSL optimizations allow reducing the secure version to $4,686 \pm 5.4\mu s$; however, the hunting and pecking process is still dominating the performance figures. For a 256-bit implementation, the OpenSSL version even showed a slight increase, requiring $17,052 \pm 22.2\mu s$.

The implementation of SPAKE2 follows RFC 9382 [23]. It uses secp256r1/secp521r1 together with the points M,N as defined in the RFC. As hash, SHA-2 with 256/512 bits digest is used. The required computation time is $1,241 \pm 1.6\mu s/1,271 \pm 1.7\mu s$ for client/server, that is, around one-fourth of the time required for Dragonfly. The 256-bit security version required $9,517 \pm 11.4\mu s/9,640 \pm 11.4\mu s$ for the client/server operations, i.e., there is only a factor of around 1.7 between Dragonfly and SPAKE2 for the higher security level. Using OpenSSL, these numbers go down to $512 \pm 0.6\mu s/537 \pm 0.6\mu s$. As for Dragonfly, the OpenSSL version was with $11,714 \pm 4.3\mu s/11,748 \pm 3.3\mu s$ slower. SPAKE2 mutually exchanges 33-byte ECC points and 16-byte authentication tags for 128-bit security and 67 byte and 32 byte for 256-bit security level.

PACE [20] was implemented with the Generic Mapping as well as the Integrated Mapping. It uses AES-CBC and PBKDF2-HMAC-SHA-2 to derive an AES key from the password. For the confirmation tag of the agreed key material, SHA-2 was used. The performance difference from the different mappings, see Table I, translates to the difference in the execution time of PACE. The runtime of the protocol using the Generic Mapping was $1,463 \pm 1.9\mu s/1,504 \pm 2.0\mu s$ on the client/server, compared to $797 \pm 1.0\mu s/856 \pm 1.1\mu s$ using the Integrated Mapping for 128-bit security level. For 256-bit security, the runtimes increase to $5,204 \pm 5.4\mu s/5,424 \pm 5.4\mu s$ and $10,477 \pm 12.8\mu s/10,736 \pm 12.7\mu s$ for PACE using the Integrated Mapping and the Generic Mapping. With the OpenSSL optimization, PACE using Generic Mapping requires $642 \pm 0.6\mu s/744 \pm 0.8\mu s$, while the version using the Integrated Mapping needs $352 \pm 0.3\mu s/401 \pm 0.4\mu s$ at a security level of 128-bits. Again, the OpenSSL version could not improve the performance at the high security level. Here, PACE with the Integrated Mapping required $5,068 \pm 2.6\mu s/5,121 \pm 2.7\mu s$ and $10,683 \pm 7.5\mu s/10,817 \pm 8.1\mu s$ for the Generic Mapping.

In addition, the Generic Mapping requires exchanging the related public keys, which is not required for the Integrated Mapping. For a security level of 128-bits, both versions send an encrypted random value of size 16+48 bytes from the client to the server and mutually exchange an EC Diffie-Hellman (ECDH) key of 33 bytes and authentication tags of 32 bytes. If the Generic Mapping is used, another ECDH key is mutually exchanged. Note that our implementation exchanges a serialized SubjectPublicKeyInfo object instead of the raw points, which increases the message size from 33 to 91 bytes. The numbers for 256-bit security increase to 16+80 bytes for the initial messages and to 67 bytes for the compressed ECC points and 64 bytes for the authentication tags. The serialized SubjectPublicKeyInfo object has a size of 158 for secp521r1.

For the implementation of CPACE [21], the Integrated Mapping was used. This scheme performs the same operations on both parties and outputs a hashed transcript and a shared point. Hence, no dedicated verification step is performed in the implementation. Avoiding the exchange of an encrypted value,

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

as is done in the PACE protocol, reduces the computation time to $715 \pm 0.7\mu s$, respectively to $352 \pm 0.3\mu s$ using OpenSSL, and the size of the exchanged messages to two times 33 bytes in the 128-bit case. For 256-bit security, the runtime of CPACE increases to $5,293 \pm 5.2\mu s$ and $5,190 \pm 2.8\mu s$ with the OpenSSL optimizations. The message size increases to two times 67 bytes in this case.

A comparison of the performance of the different balanced schemes for a security level of 128-bits is shown in Figure 6. Due to their balanced nature, the computational effort for both parties is comparable. It should be noted that the OCAKE post-quantum scheme that uses KEM is very fast compared to the other schemes at the cost of larger exchanged messages. For ECC-based schemes, the performance of optimized versions is given for comparison. Using OpenSSL provides a significant improvement for all implemented schemes using ECC for a security level of 128-bit. A similar behavior for the higher security level of 256-bit was not observed. In some cases, the OpenSSL version was even slower.

In contrast to balanced schemes, where both parties use the password, augmented schemes require a registration phase to construct a verification value that is stored on the server side.

The implementation of the augmented PAKE scheme SRP uses the 3072-bit MODP Group defined in RFC 3526 [28], since a direct translation to elliptic curves is not possible. The measured implementation follows the specification of RFC 2945 [24]. The RFC specifies the use of SHA-1. In order to follow the specification and allow for a comparable result, an instance using SHA-1 and an instance using SHA-2 (256 digest) were considered. Furthermore, a 256-bit security level instance using an 8192-bit MODP Group and SHA-2 (512 bit digest) was measured. The results show that the registration process for the SHA-2 (256) variant takes about a third longer, $885 \pm 1.3\mu s$ compared to $1,148 \pm 1.4\mu s$, while during the authentication phase, there is only a slight difference, $25,117 \pm 9.3\mu s$/$16,369 \pm 4.4\mu s$ for SHA-1 compared to $25,376 \pm 9.1\mu s$/$16,438 \pm 3.8\mu s$ for SHA-2 on the client/server. The 256-bit security level version took $12,387 \pm 27.3\mu s$, for the registration and $451,544 \pm 88.8\mu s$/$294,324 \pm 18.2\mu s$ for the client/server operations. In terms of message sizes, SRP requires the storage of a salt value, in our implementation 9 bytes and a group element of 385 bytes for the 3072-bit MODP group and 1024 bytes for the 8192-bit MODP group. The parties mutually exchange group elements of 385/1024-byte elements. In addition, the server shares the salt value with the client. The verification step consists of two proof values, one sent by the client and one sent by the server. For the SHA-1 variant, these proof values are 20 bytes each. When switching from SHA-1 to SHA-2 (256-bit digest), the size of the messages in the verification step increases from 20 to 32 byte and to 64 bytes when using a 512 bit digest.

The implementation of SPAKE2+ makes use of elliptic curves secp256r1/secp521r1, together with SHA-2 (256/512 bit output) and an HMAC key derivation function. The registration phase uses the memory-hard hash function scrypt with parameters $(32768, 8, 1)$ as recommended in the RFC. This protects

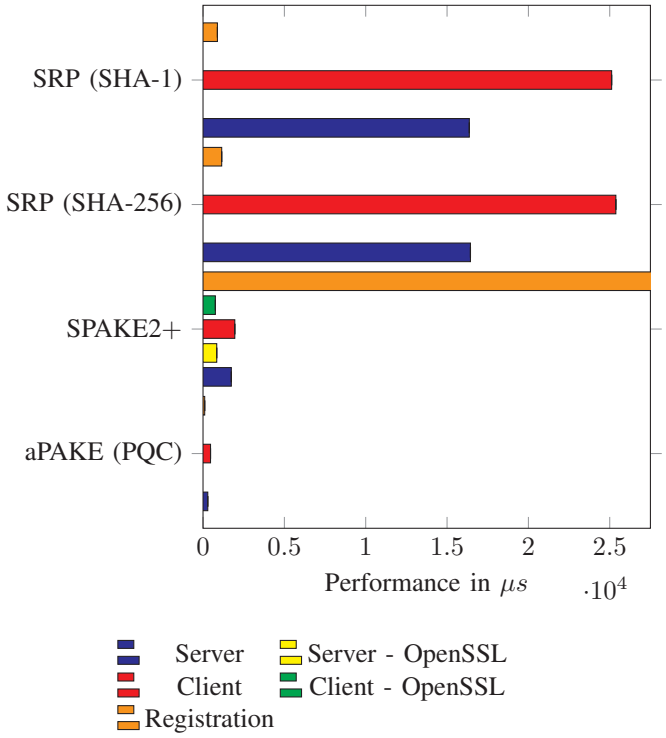


Figure 7. Performance comparison of the augmented PAKE schemes for 128-bit security level.

against offline dictionary attacks, but also leads to a significant effort during registration ($115,966 \pm 87.2\mu s$). Hence, Figure 7 does not show the full bar of the registration phase. The scrypt parameters have a direct impact on the performance of this first phase. Since the implementation relies on elliptic curves, it is, with $1,953 \pm 6.1\mu s$/$1,740 \pm 6.5\mu s$ on the client/server, faster than SRP in 128-bit variant. The registration process of the 256-bit variant took $246,155 \pm 139.6\mu s$. The client/server process required $14,128 \pm 12.5\mu s$/$12,139 \pm 10.9\mu s$. SPAKE2+ is also the only augmented scheme during our evaluation that relies on ECC. With the OpenSSL optimizations, the client/server operations could be reduced to $759 \pm 0.9\mu s$/$843 \pm 0.9\mu s$ for 128-bit security, while for the initialization phase, the memory-hard hash function dominates the performance. For 256-bit, the performance even decreases when using OpenSSL, resulting in $18,744 \pm 9\mu s$/$16,961 \pm 5.4\mu s$. Note that the client measurement does not include the application of the scrypt function, which could be done during a preparation phase/the registration. Its execution would add an effort comparable to the server registration phase. In the 128-bit security level version, the value that is stored on the server side for verification consists of a 33-byte compressed ECC point and a 32-byte value that is computed modulo the order of the base point. For the key agreement, ECC points are mutually exchanged, 33 bytes each, while verification uses 32-byte HMAC values on the client and on the server side. The 256-bit security level version stores 67+64 bytes and exchanges 67+64 byte messages mutually.

The transformation published by Lyu et al. [12] was applied

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

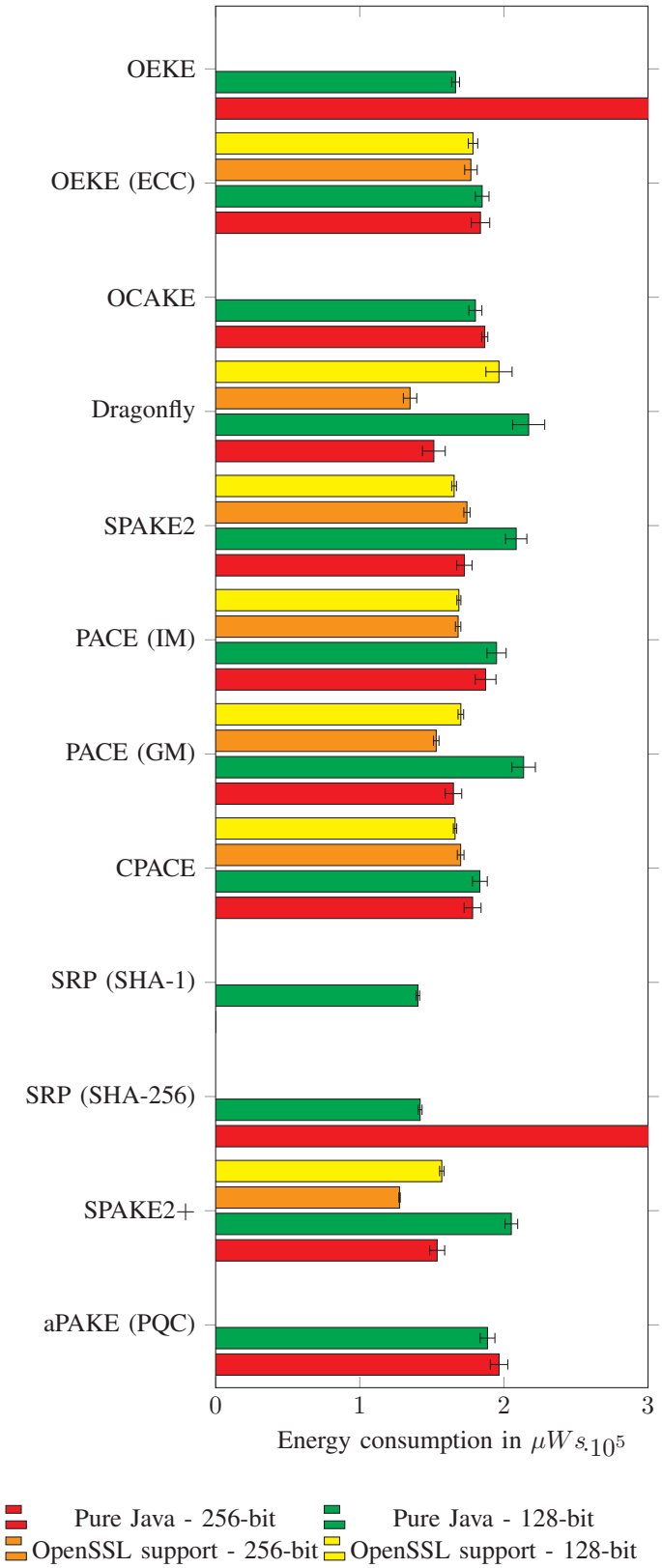


Figure 8. Energy consumption of the implemented schemes on the big CPU power rail with a 95% confidence interval

to the implemented OCAKE to compare augmented PAKE schemes with a quantum-secure scheme. In addition to ML-KEM512/ML-KEM1024, AES-Galois/Counter Mode (GCM) was used for the additional implementations, while the OCAKE protocol still employs AES-CBC. The implemented scheme outperforms the others, requiring $103 \pm 0.1\mu s/186 \pm 0.1\mu s$ for registration, $461 \pm 0.6\mu s/875 \pm 0.9\mu s$ on the client, and $292 \pm 0.5\mu s/521 \pm 0.1\mu s$ on the server for a 128/256-bit security level respectively. As it employs the balanced PQC PAKE scheme OCAKE, the message sizes are larger compared to the other schemes. The stored record requires a 32/64-byte hash value and an 800/1568-byte public key. The exchanged messages include those required for OCAKE (16+816+768/16+1584+1568) plus an encrypted key encapsulation of 16+784/16+1584 bytes for a 128/256-bit security level. The verification message uses two hash values of 32/64-byte each. It is called *aPAKE (PQC)* in Figure 7, where the performance figures of the different augmented schemes are shown.

In Figure 9 an overview of the different ECC-based schemes and their relation in terms of performance vs messages is given. The runtime given in this figure is the addition of server and client operation runtimes. The registration phase for SPAKE+ is not included in the runtime, as it is only a one-time process. SRP and OEKE that are using a discrete logarithm group are outliers in terms of required runtime and the ML-KEM based schemes OCAKE and its transformation into an augmented scheme are outliers in terms of message size. The figure shows that the difference between a 128-bit security level and a 256-bit security level, in terms of runtime, is a factor between 6.5 and 7.6. Only the runtime of Dragonfly increased by a factor of just 3.2, as its performance is driven by the mapping scheme. The message sizes of all schemes increase by a factor of around 2. While this holds also for the PQC schemes, the message sizes of the schemes based on a discrete logarithm group increase by a factor of around 2.6.

The energy consumption of the big CPU power rail is given in Figure 8. The registration phase of the augmented schemes is not included in these figures. For SRP and the PQC-secure PAKE, the registration has no significant impact on the power consumption, but for SPAKE2+, it adds $454,955 \pm 6,5062\mu Ws/458,394 \pm 6,032\mu Ws$ for the 128/256-bit security version. Hence, also in terms of power consumption is the scrypt function the dominating factor. The figure shows that, when considering a whole run of a scheme, OCAKE and its transformation into an augmented version require the same order of magnitude of energy as the other schemes for the client/server process. An outlier in this regard is SRP, which does not use ECC and hence does not require the Bouncy Castle library, but only native Java implementations. The power rails of the other CPUs, i.e., mid and little, show significantly smaller consumption, but the distribution is comparable, e.g., on the mid CPU, Dragonfly consumes with $12,669 \pm 1107\ \mu Ws$ the most energy among the schemes for a 128-bit security level, while for the 256-bit security level, SRP with $12,537 \pm 3,645\mu Ws$ consumes the most energy. The figures also show that the

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

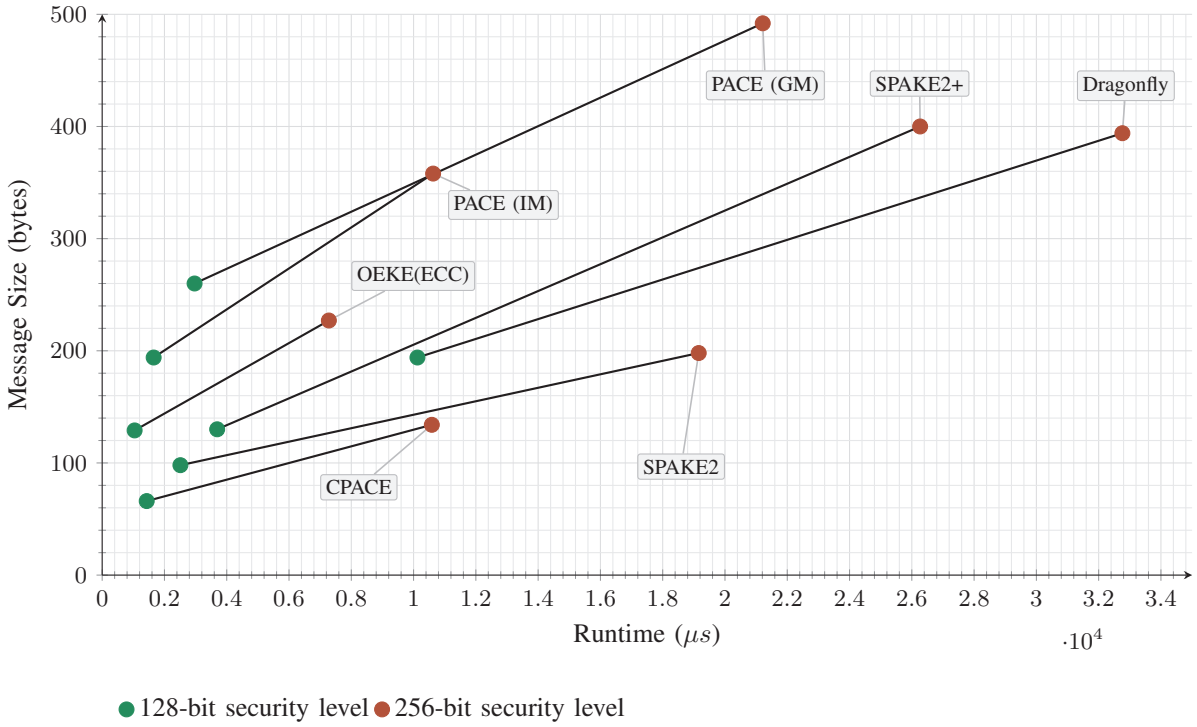


Figure 9. Relation between transferred bytes and runtime of the ECC-based PAKE schemes.

consumption for the implementations providing 256-bit security is of the same order of magnitude or even lower, despite the longer runtime. Only SRP and OEKE, which use a finite field require significantly more energy in the 256-bit versions.

An overview of all results, i.e., message sizes, runtime, power consumption of pure Java implementations and those using OpenSSL optimizations for a security level of 128 is given in Table IV and for a level of 256-bits in Table V.

However, the measured timings only consider computational effort. In order to include the network impact as well, we model its contribution as

$$T_{\text{net}} = F \cdot \frac{RTT}{2} + \frac{8B}{BW}, \tag{1}$$

where F is the number of directional online message flights, B is the total number of transmitted bytes, and BW is the available bandwidth. We count one directional message flight as one online protocol message sent from one party to the other, i.e., either client-to-server or server-to-client. Thus, two sequential directional message flights correspond approximately to one network round trip. The time a round trip requires, i.e., how long it takes for a message to travel from one endpoint to the other and for a response to come back, is denoted as RTT. Transport-layer effects, retransmissions, and packetization overhead are not part of this simplified model, but the model captures the dominant first-order latency factors.

For typical Wi-Fi or cellular bandwidths, the serialization delay of the evaluated PAKE messages is small compared to the RTT. For instance, transmitting 2,464 bytes, corresponding to the 128-bit aPAKE (PQC) instance, takes approximately 2 ms at 10 Mbit/s and less than 0.2 ms at 100 Mbit/s. Consequently, for common mobile-network settings, the number of sequential online message flights is the dominant network-dependent factor. Larger post-quantum messages become more relevant mainly on low-bandwidth or highly constrained links.

This is also reflected in Table III. It shows that, for most evaluated schemes, the number of directional online messages ranges between two and four, with PACE using Generic Mapping requiring more messages. Since each additional directional message contributes approximately $RTT/2$ to the end-to-end latency, the message count can dominate the measured cryptographic computation time in wide-area or cellular deployments. The result of combining the estimated network contribution and the computation time is given in Figure 10

V. Conclusion and Future Work

This study provides an implementation-based comparison of classical and post-quantum PAKE schemes on a modern Android device, highlighting how protocol structure, mapping functions, and cryptographic primitives influence performance. Elliptic curve–based schemes consistently outperform those based on modular arithmetic, while integrated mapping techniques significantly reduce overhead compared to iterative or interactive mappings.

Among the evaluated protocols, post-quantum candidates, such as OCAKE and aPAKE demonstrate strong computational efficiency, achieving sub-millisecond runtimes in several evaluated configurations even without hardware acceleration.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE III
ONLINE COMMUNICATION STRUCTURE OF THE EVALUATED PAKE SCHEMES. REGISTRATION FOR AUGMENTED SCHEMES IS EXCLUDED. THE LATENCY COLUMNS SHOW ONLY THE RTT-DEPENDENT CONTRIBUTION $F \cdot RTT/2$ AND OMIT THE SERIALIZATION TERM $8B/BW$.

Scheme	Online message size		Online message flights	RTT contribution in ms		
	128-bit	256-bit		RTT=5 ms	RTT=30 ms	RTT=100 ms
OEKE	832	2,144	3	7.5	45	150
OEKE (ECC)	129	227	3	7.5	45	150
OCAKE	1,664	3,296	3	7.5	45	150
Dragonfly	194	394	4	10	60	200
SPAKE2	98	198	4	10	60	200
PACE (IM)	194	358	4	10	60	200
PACE (GM)	260	492	6	15	90	300
CPACE	66	134	2	5	30	100
SRP (SHA-1)	819	–	4	10	60	200
SRP (SHA-2)	843	2,185	4	10	60	200
SPAKE2+	130	262	4	10	60	200
aPAKE (PQC)	2,464	4,896	3	7.5	45	150

Although these schemes incur higher communication overhead, they show that quantum-secure PAKEs are practical for mobile environments. In the augmented setting, the results also emphasize the trade-offs introduced by memory-hard hash functions and the potential for optimization through parameter tuning.

Our current dataset is limited to a single flagship device using the Bouncy Castle library. Generalizing absolute runtimes across the Android ecosystem requires care, because SoCs differ substantially, for example, in CPU microarchitecture, available instruction set extensions and memory hierarchy. In order to show the impact of such CPU-specific optimizations, instances using OpenSSL ECC scalar multiplication were compared to those using a Bouncy Castle implementation of the multiplication. As expected, they outperformed the pure Java implementations. The paper also presents the results for 128-bit security level implementations and for implementations providing 256-bit level. It turned out that, while the runtime required for the 256-bit versions increased, the power consumption for all but the schemes relying on finite fields remained in a similar order of magnitude. Future work will explore the impact of different SoCs, energy profiling under typical usage scenarios, and integration into complete authentication stacks. In addition, a comprehensive security and side-channel resilience evaluation will complement the performance perspective established here.

REFERENCES

[1] J.-M. Schmidt and A. Lawall, "Comparison of Password-Authenticated Key Exchange Schemes on Android", in *The Nineteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE) 2025*, Barcelona, Spain, 2025, pp. 55–61, ISBN: 978-1-68558-306-4.

[2] "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", *IEEE Std 802.11-2024 (Revision of IEEE Std 802.11-2020)*, pp. 1–5956, 2025. DOI: 10.1109/IEEESTD.2025.10979691.

[3] Connectivity Standards Alliance, *Matter Specification - Version 1.4*, Nov. 2004.

[4] R. Fillion, *Secure Remote Password (SRP): How 1Password uses it*, https://blog.1password.com/developers-how-we-use-srp-and-you-can-too/, retrieved: May, 2026.

[5] G. T. Davies *et al.*, "Security analysis of the whatsapp end-to-end encrypted backup protocol", in *Advances in Cryptology – CRYPTO 2023*, H. Handschuh and A. Lysyanskaya, Eds., Cham: Springer Nature Switzerland, 2023, pp. 330–361, ISBN: 978-3-031-38551-3.

[6] Meta, *The labyrinth encrypted message storage protocol*, https://engineering.fb.com/wp-content/uploads/2023/12/TheLabyrinthEncryptedMessageStorageProtocol_12-6-2023.pdf, retrieved: May, 2026.

[7] Apple, *Apple Platform Security - HomeKit communication security*, https://support.apple.com/en-gb/guide/security/sec3a881ccb1/web, retrieved: May, 2026.

[8] Apple, *Apple Platform Security - Escrow security for iCloud Keychain*, https://support.apple.com/en-gb/guide/security/sec3e341e75d/web, retrieved: May, 2026.

[9] Apple, *Apple Platform Security - Car key security in iOS*, https://support.apple.com/en-gb/guide/security/secf64471c16/web, retrieved: May, 2026.

[10] H. Beguinet, C. Chevalier, D. Pointcheval, T. Ricosset, and M. Rossi, "Get a cake: Generic transformations from key encapsulation mechanisms to password authenticated key exchanges", in *Applied Cryptography and Network Security*, M. Tibouchi and X. Wang, Eds., Cham: Springer Nature Switzerland, 2023, pp. 516–538, ISBN: 978-3-031-33491-7.

[11] N. Alnahawi *et al.*, *Post-quantum cryptography in eMRTDs: Evaluating PAKE and PKI for travel documents*, Cryptology ePrint Archive, Paper 2025/812, 2025.

[12] Y. Lyu, S. Liu, and S. Han, "Efficient Asymmetric PAKE Compiler from KEM and AE", in *Advances in Cryptology – ASIACRYPT 2024: 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9–13, 2024, Proceedings. Part V*, Kolkata, India: Springer-Verlag, 2024, pp. 34–65. DOI: 10.1007/978-981-96-0935-2_2.

[13] N. Alnahawi, D. Haas, E. Mauß, and A. Wiesmaier, *SoK: PQC PAKEs - cryptographic primitives, design and security*, Cryptology ePrint Archive, Paper 2025/119, 2025.

[14] D. Taylor, T. Perrin, T. Wu, and N. Mavrogiannopoulos, *Using the Secure Remote Password (SRP) Protocol for TLS Authentication*, RFC 5054, Nov. 2007. DOI: 10.17487/RFC5054.

[15] J.-M. Schmidt, *Requirements for Password-Authenticated Key Agreement (PAKE) Schemes*, RFC 8125, Apr. 2017. DOI: 10.17487/RFC8125.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

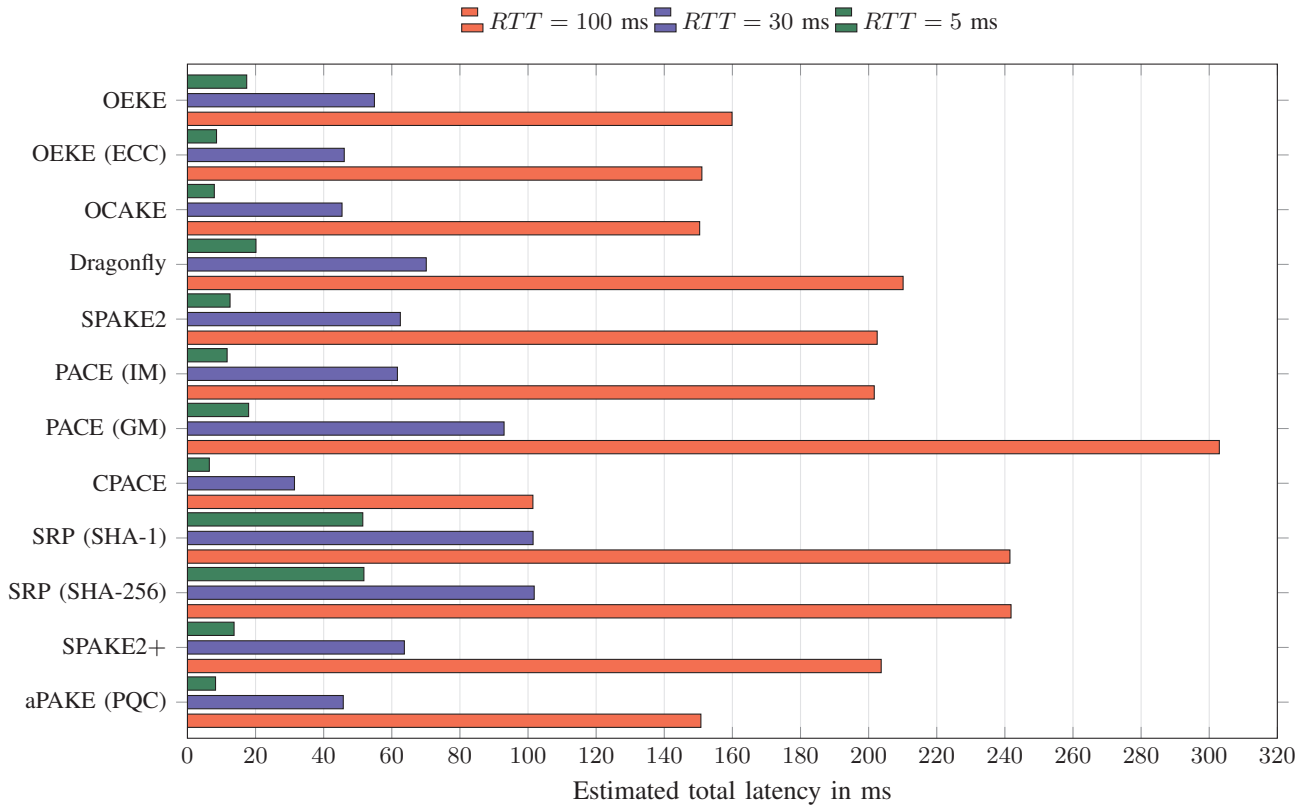


Figure 10. Estimated end-to-end latency of the evaluated PAKE schemes for the 128-bit security level using the pure Java implementation. The values combine measured total client/server computation time with the modeled network contribution $F \cdot RTT/2$. Registration of augmented schemes is excluded; serialization delay is omitted.

TABLE IV
OVERVIEW OF THE IMPLEMENTATION RESULTS FOR A 128-BIT SECURITY LEVEL

Scheme	Message Bytes	Pure Java Time μs	Pure Java Energy μWs	OpenSSL support Time μs	OpenSSL support EnergyμWs
Balanced Schemes					
OEKE	832	9,903±1.9	166,505±2,706	-	-
OEKE (ECC)	129	1,042±0.9	184,908±4,748	658±0.4	178,569±3,283
OCAKE	1,664	393±0.3	180,235±4,497	-	-
Dragonfly	194	10,124±9.6	217,196±11,116	9,372±7.6	196,597±9,087
SPAKE2	98	2,512±2.4	208,527±7,402	1,049±0.8	165,464±1,650
PACE (IM)	194	1,653±1.5	194,880±6,669	753±0.5	168,750±1,382
PACE (GM)	260	2,967±2.8	213,672±8,168	1,386±1.0	170,110±1,936
CPACE	66	1,430±1.0	183,313±5,158	558±0.4	166,107±1,165
Augmented Schemes					
SRP (SHA-1)	819	41,486±10.3	140,339±1,233	-	-
SRP (SHA-2)	843	41,814±9.9	141,873±1,352	-	-
SPAKE2+	130	3,693±8.9	205,158±4,400	1,602±0.9	156,919±1,559
aPAKE (PQC)	2,464	753±0.8	188,650±5,128	-	-

[16] S. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks", in *Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, 1992, pp. 72–84. DOI: 10.1109/RISP.1992.213269.

[17] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks", in *Advances in Cryptology — EUROCRYPT 2000*, B. Preneel, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 139–155, ISBN: 978-3-540-45539-4.

[18] E. Bresson, O. Chevassut, and D. Pointcheval, "Security proofs for an efficient password-based key exchange", in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, ser. CCS '03, Washington D.C., USA: Association for Computing Machinery, 2003, pp. 241–250, ISBN: 1581137389. DOI: 10.1145/948109.948142.

[19] D. Harkins, *Dragonfly Key Exchange*, RFC 7664, Nov. 2015. DOI: 10.17487/RFC7664.

[20] ICAO - International Civil Aviation Organization, *Doc 9303 - machine readable travel documents- part 11: Security mecha-*

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE V
OVERVIEW OF THE IMPLEMENTATION RESULTS FOR A 256-BIT SECURITY LEVEL

Scheme	Message Bytes	Pure Java Time μs	Pure Java Energy μWs	OpenSSL support Time μs	OpenSSL support EnergyμWs
Balanced Schemes					
OEKE	2,144	586,732±30.3	1,084,009±6,287	-	-
OEKE (ECC)	227	7,280±5.7	183,733±6,422	7,377±5.1	177,112±4,352
OCAKE	3,296	788±0.5	186,710±6,057	-	-
Dragonfly	394	32,762±31.5	151,394±7,902	34,104±31.4	134,905±4,649
SPAKE2	198	19,157±16.1	172,604±5,310	23,462±5.4	174,350±2,158
PACE (IM)	358	10,628±7.6	187,330±7,165	10,189±3.7	168,235±1,862
PACE (GM)	492	21,213±18.0	164,984±5,737	21,500±11.0	153,044±1,909
CPACE	134	10,586±7.6	178,290±5,863	10,380±4.0	169,997±2,304
Augmented Schemes					
SRP (SHA-2)	2,185	745,868±90.6	1,417,541±5,006	-	-
SPAKE2+	262	26,267±16.6	153,709±5,195	35,705±10.5	127,600±492
aPAKE (PQC)	4,896	1,396±0.9	196,660±6,057	-	-

nisms for mrtds, eighth edition, https://www.icao.int/publications/Documents/9303_p11_cons_en.pdf, 2021.

[21] M. Abdalla, B. Haase, and J. Hesse, "CPace, a balanced composable PAKE", Internet Engineering Task Force, Internet-Draft draft-irtf-cfrg-cpace-14, Apr. 2025, Work in Progress, 96 pp.

[22] A. Faz-Hernandez, S. Scott, N. Sullivan, R. S. Wahby, and C. A. Wood, *Hashing to Elliptic Curves*, RFC 9380, Aug. 2023. DOI: 10.17487/RFC9380.

[23] W. Ladd, *SPAKE2, a Password-Authenticated Key Exchange*, RFC 9382, Sep. 2023. DOI: 10.17487/RFC9382.

[24] T. Wu, *The SRP Authentication and Key Exchange System*, RFC 2945, Sep. 2000. DOI: 10.17487/RFC2945.

[25] T. Taubert and C. A. Wood, *SPAKE2+, an Augmented Password-Authenticated Key Exchange (PAKE) Protocol*, RFC 9383, Sep. 2023. DOI: 10.17487/RFC9383.

[26] Android Developers, *Microbenchmark*, https://developer.android.com/topic/performance/benchmarking/microbenchmark-overview, retrieved: May, 2026.

[27] Perfetto, *System profiling, app tracing and trace analysis*, https://perfetto.dev/, retrieved: May, 2026.

[28] M. Kojo and T. Kivinen, *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)*, RFC 3526, May 2003. DOI: 10.17487/RFC3526.

[29] T. Polk, R. Housley, S. Turner, D. R. L. Brown, and K. Yiu, *Elliptic Curve Cryptography Subject Public Key Information*, RFC 5480, Mar. 2009. DOI: 10.17487/RFC5480.

[30] NIST, "Module-lattice-based key-encapsulation mechanism standard", U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publication (FIPS) 203, 2024. DOI: 10.6028/NIST.FIPS.203.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Beyond Prevalence: Quantifying the Behavioural Effectiveness of Persuasion Principles in Phishing Attacks

Alexander Lawall
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
e-mail: alexander.lawall@iu.org

***Abstract*—This paper presents a mixed-method investigation into how psychological persuasion mechanisms translate into behavioural effectiveness in phishing attacks, with a specific focus on Cialdini's six principles of influence. A qualitative analysis of authentic spear-phishing emails is complemented by a quantitative study of 300 phishing samples spanning ten attack types to address three research questions. Beyond descriptive frequency analysis, the study employs a multi-stage statistical framework combining linear regression, binomial logistic regression, interaction modelling, and unsupervised clustering. The findings show that while *scarcity* remains the most frequently used persuasion tactic, it does not significantly predict user compromise across models. Instead, *liking* and *authority* consistently emerge as the most effective predictors of phishing success, with interaction effects indicating that composite persuasion strategies further amplify attack effectiveness. Clustering analysis reveals modality-dependent persuasion profiles across phishing attack types, highlighting structural differences between high- and low-bandwidth attack channels. These results demonstrate a mismatch between commonly deployed and behaviourally potent influence strategies and provide empirical evidence for the strategic layering of persuasion in phishing. The study offers implications for user awareness training, Natural Language Processing (NLP)-enhanced detection, and the design of psychologically informed defense mechanisms.**

***Keywords*-Phishing; Social Engineering; Psychological Persuasion; Cialdini's Influence Principles; Behavioral Cybersecurity.**

I. Introduction

A. Motivation and Background

This work is an extended version of *Quantifying Persuasion – A Comparative Analysis of Cialdini's Principles in Phishing Attacks*, published at SECURWARE 2025 [1]. Phishing has remained one of the most prevalent and financially damaging forms of cybercrime since its emergence in the 1990s [2]. Despite continuous advancements in technical countermeasures, attackers consistently exploit the human element, which remains the weakest link in cybersecurity. Recent statistics show that up to 80% of security breaches are attributed to human error, underscoring the need for behavioural and psychological countermeasures alongside technical controls [3].

Current phishing campaigns frequently use psychological manipulation rather than exploiting technical vulnerabilities [4] [5]. Specifically, attackers embed persuasive elements within their messages to increase credibility [6]. Among the most robust frameworks for analyzing these manipulations are the six principles of social influence developed by Robert Cialdini: *Reciprocity*, *Liking*, *Social Proof*, *Authority*, *Scarcity*, and *Commitment/Consistency* [7] [8] [9]. These principles have been widely adopted by attackers in phishing, spear-phishing, and vishing campaigns, making them critical to understanding adversarial social engineering.

B. Research Objectives and Questions

This study aims to investigate how psychological persuasion, particularly Cialdini's six classical principles, is operationalized in phishing attacks, and to determine which principles are most strongly associated with successful compromise. Prior work in this area either focuses on case-based interpretations of real and hypothetical phishing emails [10] [11], or applies statistical modeling to large corpora of phishing incidents [12] [13]. However, there is a lack of research that integrates both qualitative and quantitative perspectives to comparatively evaluate the psychological mechanics behind phishing efficacy.

The following research questions address this gap:

RQ1 *"How are Cialdini's six classical principles of influence exploited in real-world phishing and spear-phishing attacks?"*

RQ2 *"What is the statistical prevalence of each principle across phishing types?"*

RQ3 *"Which principles are most strongly associated with victim compromise, and why?"*

C. Contribution and Structure

This paper contributes a mixed-methods analysis of persuasion in phishing attacks by (1) synthesizing how Cialdini's six principles manifest in phishing attacks, (2) quantifying their intensity and frequency across multiple phishing modalities, and (3) modeling their predictive power for user compromise using the statistical relationship between the application principle and the success of the phishing. The findings highlight a critical gap between commonly used tactics (e.g., scarcity) and the most behaviorally effective ones (liking and authority), offering implications for awareness training, Natural Language Processing (NLP)-based detection, and psychologically informed defenses.

The remainder of the paper is structured as follows. Section II reviews related work on persuasion in phishing and positions this contribution within existing literature. Section III introduces the theoretical background on Cialdini's framework and phishing typologies. Section IV describes the qualitative

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

and quantitative methods used. Section V presents the empirical results, while Section VI discusses implications for cybersecurity practice. Section VII concludes with a summary and outlook for future research.

II. Related Work

Research on phishing attacks highlights the role of social engineering and psychological manipulation as key drivers of victim compromise. Among the most widely used frameworks for examining these tactics are Cialdini's six principles of influence. Prior work demonstrates that these principles are systematically exploited across diverse phishing modalities, yet their behavioral potency varies considerably.

Content analyses and simulations consistently find *authority* and *social proof* to be the most prevalent principles. Taib et al. [14] conducted a meta-analysis of over 56,000 participants and 81 studies, showing that authority-based manipulations not only dominate phishing messages but also lead to significantly higher compromise rates. Similarly, Ahmad et al. [15] found that man-in-the-middle phishing attacks employ social proof in 76.1% of cases and liking in 74.6%, illustrating the systematic use of group conformity and familiarity cues. In spear-phishing, Uehara et al. [16] documented authority usage rates as high as 96.1% and scarcity in 41.1% of attacks, highlighting the tailoring of principles to hierarchical and urgent contexts. By contrast, reciprocity (16.4%) and commitment/consistency (1–2%) remain less common, though their use is increasing over time in certain contexts [6].

While principle prevalence is well documented, fewer studies directly assess their effectiveness in predicting compromise. Experimental research shows authority consistently yields high success rates. Bona and Paci [17] report a 21.5% compromise rate in organizational phishing exercises driven by authority cues, while Butavicius et al. [18] confirm its dominance across spear-phishing contexts. Social proof also emerges as a strong predictor, particularly in finance and public sector settings where conformity to peer behavior or industry norms is salient [19]. Liking is less frequently used in generic phishing but has proven highly effective in personalized contexts such as vishing or social media-based attacks, where rapport and similarity cues are stronger. Scarcity shows mixed results: although frequent, it may suffer from diminishing returns due to user desensitization in environments saturated with urgency cues [6] [20]. Reciprocity appears context-dependent, being more effective among older adults [21], but in some cases correlates negatively with compromise likelihood, possibly reflecting heightened awareness of unsolicited "favors".

The literature also shows contextual and demographic moderators shaping susceptibility. Lawson et al. [22] and subsequent studies suggest that personality traits interact with persuasion tactics, while age is a strong predictor of susceptibility to reciprocity-based influence [21]. Organizational culture and industry also affect outcomes: Tian et al. [19] demonstrate that authority cues are especially effective in finance and public administration, whereas liking is more influential in non-financial contexts. Furthermore, attack modality influences principle application, i.e., scarcity dominates in low-bandwidth channels like SMS, while liking and commitment gain prominence in richer contexts such as spear-phishing and vishing.

More recent research indicates the evolving nature of phishing. AI-generated phishing campaigns increasingly combine multiple principles, blending emotional tone with contextual realism [23] [24]. Longitudinal analyses reveal that while reciprocity and social proof are declining in prevalence, scarcity and commitment/consistency are on the rise, suggesting attacker adaptation to user awareness over time [6]. Hybrid strategies that combine principles, such as authority with scarcity or liking with social proof, have been shown to produce synergistic effects [25].

Despite extensive empirical work, existing studies often focus either on prevalence (content analysis) or effectiveness (experiments and field tests), but rarely integrate both perspectives. Moreover, few studies systematically compare principles across diverse phishing modalities while simultaneously modeling their predictive power for compromise. This creates a critical gap between descriptive and causal insights. Addressing this gap, this paper contributes a mixed-method approach, combining qualitative analysis of authentic phishing emails with quantitative modeling of 300 samples across ten attack types. This integration enables a more nuanced assessment of both the strategic deployment and behavioral impact of Cialdini's six classical principles in phishing attacks.

III. Theoretical Foundation

A. Cialdini's six Classical Principles of Influence

Robert Cialdini's theory of persuasion outlines six core psychological principles that shape human decision-making and compliance [7]. These principles are frequently exploited in phishing campaigns and form the analytical backbone of this study.

1) **Reciprocity:** People feel obligated to return a favor, even if unsolicited [26]. In phishing, this manifests through fake assistance, rewards, or account-support scenarios that pressure the victim into "returning" the perceived help. For example, attackers may send messages such as "We detected suspicious activity and restored your account access. Please confirm your credentials to complete the protection process." The victim is psychologically encouraged to reciprocate the alleged support by complying with the request.
2) **Liking:** Users are more likely to comply with requests from individuals or brands they find likable, familiar, or socially close. Attackers exploit this by impersonating colleagues, friends, supervisors, or trusted brands to reduce suspicion and establish rapport [27]. A common example is a phishing email appearing to come from a friendly coworker stating: "Hi Tom, great work on the report. Could you quickly review this shared document for me?". Such familiarity increases trust and lowers critical evaluation. This principle strongly correlates with compromise likelihood [14].

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

3) **Social Proof:** Individuals tend to follow behaviors exhibited by others, especially in uncertain situations. Phishing messages exploit this tendency by implying that peers, coworkers, or other users have already complied with the requested action [28]. For example, attackers may write: "Most employees in your department have already completed the mandatory security verification." This creates perceived normative pressure and increases the likelihood of compliance.
4) **Authority:** Compliance increases when messages appear to originate from legitimate authority figures. This is one of the dominant principles in spear-phishing, CEO fraud, and Business Email Compromise attacks, where attackers impersonate executives, IT administrators, banks, or government institutions [18]. For instance, an attacker may send an urgent email claiming: "This is the CFO. Please process the attached payment request before 3 p.m.". The perceived hierarchical legitimacy discourages verification and encourages immediate action.
5) **Scarcity:** Limited-time opportunities or threats of loss trigger urgency and reduce reflective decision-making. Phishing emails frequently exploit scarcity through warnings of account suspension, expired access, or urgent deadlines [6]. A typical example is: "Your account will be suspended within 30 minutes unless you verify your login credentials immediately." Such urgency pressures users into acting before critically evaluating the message.
6) **Commitment and Consistency:** Once individuals commit to a small action, they are more likely to continue with larger requests in order to remain behaviorally consistent [29]. Phishing attacks often exploit this principle through staged interactions that escalate gradually [30]. For example, a victim may first be asked to confirm a harmless notification, then log into a portal, and later approve a transaction or disclose credentials. Each prior action psychologically reinforces continued compliance.

A later extension of Cialdini's framework introduces **Unity** as an additional principle of influence [31]. Unity refers to persuasion based on shared identity, belonging, and perceived membership in the same group. In phishing, Unity may appear when attackers frame the message as coming from "one of us", such as a colleague from the same department, a member of the same university, a fellow employee, a professional association, or a shared national, religious, or organizational community. For example, a phishing email may state, "As members of the security team, we all need to complete this verification today", thereby activating a sense of collective identity rather than merely authority or social proof. Unity is conceptually related to liking and social proof but differs because the persuasive force comes from shared identity and in-group belonging.

The present study focuses on the six classical principles because the coding framework, prior comparative literature, and regression models were designed around those six constructs. However, Unity is particularly relevant for spear-phishing, BEC, whaling, and AI-generated phishing, where attackers can personalize messages using organizational or social identity cues.

These principles are not mutually exclusive and are often combined strategically in phishing scenarios [32]. They represent well-documented psychological heuristics that adversaries exploit to bypass cognitive defenses.

B. Social Engineering and Phishing Taxonomy

Social Engineering refers to the manipulation of human behavior to gain unauthorized access or extract confidential data. Unlike technical exploits, social engineering targets cognitive biases and emotional responses [32] [6].

Phishing, a subclass of social engineering, takes multiple forms depending on delivery method, personalization, and attacker intent [33]. The following taxonomy outlines ten studied phishing/attack types:

1) **Generic Phishing:** Broad, untargeted campaigns often impersonating major services (e.g., banks, delivery services). These rely on volume and simple cues like logos or time-sensitive warnings [34].
2) **Spear-Phishing:** Tailored attacks on individuals, typically leveraging Open-Source Intelligence (OSINT) to personalize content [35]. Spear-phishing has high success rates due to contextual plausibility [36].
3) **Business Email Compromise (BEC):** A subtype of spear-phishing where attackers impersonate executives to fraudulently initiate financial transactions [37]. Authority and urgency dominate these attacks.
4) **CEO-Fraud:** A further specialization of BEC in which attackers spoof high-level executives to manipulate subordinates into performing unauthorized tasks, often financial [38]. Strongly driven by authority and obedience heuristics.
5) **Whaling:** A form of spear-phishing targeting high-profile individuals such as C-level executives or board members [39]. These attacks combine authority with high contextual relevance, often mimicking internal workflows.
6) **Clone-/Dynamite-Phishing:** Involves copying legitimate past communications and resending them with malicious payloads or links. This method exploits trust in established communication patterns and prior context.
7) **Vishing:** Voice-based phishing via phone calls. Attackers impersonate authorities or support personnel [40]. Vishing exploits real-time pressure, often employing the commitment and authority principles.
8) **Quishing:** QR-code phishing attacks that exploit users' trust in QR-based scenarios. Quishing bypasses URL verification and often embeds commitment through routine-seeming steps [41].
9) **Smishing:** SMS-based phishing that mimics alerts from banks, couriers, or apps. Its scarcity and urgency lead to quick, unreflective responses [42].

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

10) **AI-Based Phishing:** Uses Large Language Models (LLM) or Generative AI to create highly convincing and personalized phishing content at scale [23]. These attacks increasingly integrate emotional tone, contextual cues, and stylistic mimicry, enhancing the persuasiveness of principles like liking and authority [24].

Human factors in cybersecurity remain critical. Attackers increasingly adapt their strategies to exploit known psychological vulnerabilities, not just technological gaps. These include cognitive overload, authority bias, time pressure, and familiarity illusions [43]. Understanding how influence principles manifest across phishing variants is essential for designing more effective awareness training and detection mechanisms.

IV. Methodology

A. Case-Based Qualitative Analysis

A qualitative case study approach was employed based on real spear-phishing examples to explore how Cialdini's influence principles are operationalized. Therefore, authentic spear-phishing emails were drawn from documented APT campaigns and original email scenarios, each designed to exemplify Cialdini's six principles. Each case was examined through critical textual analysis, focusing on linguistic markers, attacker strategy, and contextual cues that demonstrated the activation of psychological triggers. The qualitative analysis aimed to answer how each principle is exploited in practice. This analysis provides both validity and conceptual diversity.

B. Quantitative Content Analysis

A content analysis was conducted to statistically assess the prevalence and intensity of the six principles across different phishing methods. The dataset comprised 300 phishing emails, evenly distributed across the ten attack types (cf. Section III-B). Each email was manually coded using a predefined scale (0-5) for the six principles. Coding followed a deductive scheme grounded in Cialdini's theory, and a structured coding guide ensured inter-case consistency. This method allowed for fine-grained measurement of psychological influence intensity and variation across modalities. Friedman tests were performed, followed by pairwise Wilcoxon signed-rank tests with Bonferroni correction to assess within-group variance and test for statistically significant differences between principles within each attack type. The Friedman test was selected because the study design involved repeated measures across the same set of phishing samples evaluated on six related persuasion principles. Unlike ANOVA, which assumes normality, the Friedman test is a non-parametric equivalent suitable for ordinal or non-normally distributed data. Following this, Wilcoxon signed-rank tests with Bonferroni correction were applied for pairwise comparisons. This choice reflects their suitability for dependent samples where measurements are related (i.e., the same phishing email coded for multiple principles) and where assumptions of parametric tests (normal distribution, homoscedasticity) are not met.

C. Regression and Statistical Evaluation

1) Baseline OLS Regression Model: A multiple linear regression model was developed using Ordinary Least Squares (OLS), with the dependent variable being the compromise rate per attack type to quantify the relationship between principles and phishing success. The independent variables were the principle intensity scores per email, resulting in a total of 300 observations with six predictors. The baseline OLS regression is specified in conceptual form as:

$$Y_i = \beta_0 + \sum_{k=1}^{6} \beta_k X_{ik}, \tag{1}$$

where Y_i denotes the observed compromise outcome (or compromise rate) for sample i, X_{ik} represents the intensity of persuasion principle k in sample i, β_0 is the intercept, β_k are the main-effect coefficients. Throughout this paper, the coefficient indices correspond to the following ordering of persuasion principles: (β_1) Reciprocity, (β_2) Liking, (β_3) Social Proof, (β_4) Authority, (β_5) Scarcity, and (β_6) Commitment/Consistency, cf. order in Table II. Although phishing outcomes are conceptually binary at the individual level (compromised vs. not compromised), the dependent variable Y_i represents an empirically assigned compromise rate associated with each phishing sample based on its attack type. This formulation enables regression analysis at the sample level while preserving comparability across attack categories.

The model was validated using standard assumption checks. Multicollinearity was tested via Variance Inflation Factors (VIF), heteroskedasticity was assessed via the Breusch-Pagan test, and normality of residuals via Q-Q plots and histograms. Due to violations of homoskedasticity and residual normality, robust HC3 standard errors were applied. Significant predictors were identified based on $p < 0.05$. Positive predictors of phishing success are confirmed via confidence intervals. In summary, non-parametric tests were employed for within-group comparisons due to the ordinal nature and non-normal distribution of principle intensity X_{ik} scores, while regression modeling with robust errors allowed us to examine predictive relationships at the continuous level, compensating for assumption violations. Together, these methods ensured both robustness and interpretability for behavioral security data.

The OLS model is retained as a transparent linear baseline that facilitates direct comparison with prior phishing studies. The GLM is subsequently introduced as a distributionally appropriate robustness check that accounts for the bounded and non-Gaussian nature of compromise probabilities.

2) Generalized Linear Model (GLM): Additional statistical models were introduced beyond the OLS regression to strengthen the inferential validity of the analysis. While OLS provides an interpretable linear baseline, phishing-success data typically exhibit non-Gaussian residual distributions, heteroskedasticity, and in several cases, non-linear interactions between psychological principles. Therefore, the modeling framework was expanded along three dimensions: (1) generalized linear modeling, (2) interaction modeling, and (3)

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

unsupervised clustering of phishing attack types. The third extension focuses on unsupervised clustering of phishing attack types to identify latent persuasion profiles across attack modalities.

Compromise rates were treated as continuous values aggregated at the attack-type level. However, from a behavioural-security standpoint, phishing outcomes are binary at the individual level (compromised vs. not compromised). A binomial logistic regression model was estimated using the principle-intensity scores from all 300 samples as predictors to approximate this structure. While the dependent variable remained aggregated by method, the logit model provides a more appropriate functional form for bounded probability outcomes.

For the GLM, the same linear predictor as in the Baseline OLS Regression Model is mapped to the binary compromise outcome via the logit link function:

$$\text{logit}(p_i) = \eta_i = \beta_0 + \sum_{k=1}^{6} \beta_k X_{ik}, \tag{2}$$

where $p_i = \Pr(Y_i = 1)$ denotes the estimated probability of compromise for sample i, and X_{ik} represents the coded intensity of influence principle k. η_i denotes the linear predictor. The intercept term β_0 denotes the baseline log-odds of compromise when all principle-intensity scores are zero, capturing the inherent susceptibility level independent of any persuasive cues. Each coefficient β_k quantifies the change in the log-odds of compromise associated with a one-unit increase in the respective influence principle k, holding all other principles constant; positive values indicate that a higher intensity of that principle increases the estimated compromise probability, whereas negative values indicate a protective effect.

3) Interaction Modelling: The OLS model treated each principle as an independent predictor. However, persuasion research suggests that attackers often employ composite influence strategies in which principles amplify one another (e.g., Authority combined with Scarcity). A series of pairwise interaction terms was added to the model to test this assumption:

$$\eta_i = \beta_0 + \sum_{k=1}^{6} \beta_k X_{ik} + \sum_{k<j} \beta_{kj} X_{ik} X_{ij}, \tag{3}$$

where the interaction terms $\beta_{kj} X_k X_j$ capture the extent to which the joint presence of two principles produces effects beyond their individual contributions. Given the 15 possible two-way interactions, Bonferroni-adjusted significance thresholds were applied to control the error rate.

4) Model Robustness and Comparative Performance: Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC) are defined as follows:

$$\text{AIC} = 2a - 2\ln(\hat{L}), \tag{4}$$

$$\text{BIC} = a\ln(n) - 2\ln(\hat{L}), \tag{5}$$

where a denotes the number of estimated parameters in the model, n is the sample size, and $\ln(\hat{L})$ represents the maximized log-likelihood of the model. Both criteria penalize excessive model complexity: AIC applies a linear penalty ($2a$), whereas BIC employs a stronger, sample-size-dependent penalty ($a\ln(n)$). For a logistic regression model, the maximized log-likelihood is computed as

$$\ln(\hat{L}) = \sum_{i=1}^{n} \left[y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i) \right], \tag{6}$$

where $p_i = \Pr(Y_i = 1)$ denotes the predicted compromise probability obtained by applying the inverse logit function to the linear predictor η_i, i.e., $p_i = \frac{1}{1+e^{-\eta_i}}$. $y_i \in \{0, 1\}$ denotes the observed binary outcome for sample i (compromised vs. not compromised). This equation aggregates the contribution of all n observations (300 in this study) and forms the basis for computing information criteria such as AIC and BIC. Lower values of AIC and BIC indicate a more efficient model, achieving a better balance between goodness-of-fit and parsimony.

D. Limitations and Ethical Considerations

The mixed-methods approach has inherent limitations. The qualitative case study relies on subjective interpretation of attacker intent and message construction, which may limit reproducibility. The quantitative analysis is limited by its reliance on public datasets (i.e., PhishTank, OpenPhish, APWG eCrime Exchange), which may underrepresent more sophisticated or covert phishing techniques.

From an ethical standpoint, the dual-use nature of this research: the insights gained into manipulation strategies could potentially be misused. However, the aim is to empower defenders to recognize and mitigate socially engineered threats. No Personally Identifiable Information (PII) was included, and all real phishing emails used were publicly disclosed by security researchers.

V. Results and Discussion

A. Patterns of Influence in Phishing

The qualitative analysis of real-world phishing emails revealed strategic and differentiated use of Cialdini's influence principles. For example, attackers exploiting *authority* frequently impersonated C-level executives or institutional leaders, incorporating formal signatures and authoritative tone to enforce compliance. A spear-phishing email targeting the Afghan National Security Council used institutional logos and urgency to simulate government hierarchy.

Liking was exploited via impersonation of familiar senders, such as colleagues or friends, while *social proof* was invoked through phrases suggesting peer compliance (e.g., "your team has already updated credentials"). The framework was further expanded through scenarios that demonstrated nuanced manipulations, such as using perceived similarity or shared values to activate *commitment and consistency*.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

B. Statistical Prevalence Across Attack Types

The quantitative analysis of 300 phishing samples (10 attack types $\times$ 30 emails each) across ten attack types revealed distinct patterns in the intensity and distribution of influence principles. Table I summarizes the median influence scores of the ten attack types. It is important to note that descriptive frequency counts alone could not establish whether observed differences across principles were statistically reliable. The Friedman and Wilcoxon tests thus provided a rigorous basis for determining whether the differences in principle intensity X_{ik} were significant rather than artifacts of sample variation. It highlights that influence principles are functionally adapted to each attack type. For example, scarcity dominates in smishing and generic phishing due to limited message length, while authority and commitment prevail in BEC, CEO-fraud, and Whaling scenarios. Thus, influence principles are selected based on attack modality, channel limitations (e.g., SMS vs. email), and attacker objectives.

TABLE I
RELEVANCE OF CIALDINI'S SIX CLASSICAL PRINCIPLES BY ATTACK TYPE (MEDIAN)

Attack Type	*Reciprocity*	*Commit./Consist.*	*Social Proof*	*Liking*	*Authority*	*Scarcity*
Generic Phishing	0.00	3.00	0.00	0.00	2.00	5.00
Spear-Phishing	0.00	3.00	0.00	0.00	3.50	5.00
BEC	0.00	4.50	0.00	1.00	4.00	5.00
CEO-Fraud	0.00	5.00	0.00	1.00	5.00	5.00
Whaling	0.00	5.00	0.00	1.00	5.00	5.00
Vishing	0.00	4.50	0.00	1.00	4.00	5.00
Clone-/Dyn.-Phish.	0.00	3.00	0.00	0.00	2.00	5.00
Quishing	0.00	2.50	0.00	0.00	2.00	4.00
Smishing	0.00	2.00	0.00	0.00	2.00	5.00
AI-based Phishing	0.00	3.00	0.00	1.00	4.00	5.00

The principle of *scarcity* was the most consistently applied, with a median intensity of 5 across the attack types. *Authority* and *commitment and consistency* were also prevalent, especially in BEC, CEO-fraud, and Whaling attacks, where hierarchical compliance and task escalation were common. Conversely, *reciprocity*, *social proof*, and *liking* were less frequently used overall, though they appeared more often in high-personalization scenarios such as vishing and AI-enhanced phishing.

C. Regression Results and Behavioural Interpretation

1) Baseline OLS Regression Model: The OLS regression model, estimated with HC3 robust standard errors, models the relationship between persuasion principle intensities X_{ik} and the observed compromise outcome Y_i. The adjusted R^2 of 0.126 indicates that approximately 12.6% of the variance in Y_i (i.e., the compromise rate) is explained by the six Cialdini principle intensity X_{ik} scores. This level of explanatory power is acceptable for behavioural-security modelling, where human decision-making introduces substantial noise.

All Variance Inflation Factors (VIF) were below 1.4, indicating negligible multicollinearity among the predictors and confirming that the six persuasion principles are statistically distinct in the dataset. This supports a reliable interpretation of the estimated coefficients.

While Table I reports descriptive prevalence and median intensity patterns, Table II quantifies the predictive contribution of each persuasion principle to compromise outcomes across the 300 phishing emails. The results demonstrate that not all principles contribute equally to phishing success. *Liking* exhibits the strongest and most statistically significant effect ($\beta_2 = 0.6030$, $p < 0.001$), supporting the hypothesis that affective closeness, familiarity, and interpersonal mimicry substantially increase user compliance. *Authority* is also a significant predictor ($\beta_4 = 0.2011$, $p = 0.018$), consistent with prior findings that impersonation of executives, IT staff, or institutional figures triggers obedience under hierarchical pressure.

In contrast, *Reciprocity* shows a statistically significant negative association with compromise likelihood ($\beta_1 = -0.0263$, $p = 0.0234$), suggesting that explicit offers of unsolicited assistance may raise suspicion rather than trust. Despite their high prevalence, *Scarcity* ($\beta_5 = 0.0118$, $p = 0.081$) and *Commitment/Consistency* ($\beta_6 = 0.0142$, $p = 0.092$) do not reach statistical significance. This discrepancy between frequency and predictive power highlights a central insight: frequent use of an influence tactic does not necessarily imply behavioural effectiveness.

Overall, the OLS results empirically ground the prioritization of *Liking* and *Authority* in phishing detection mechanisms and user awareness training, while suggesting diminishing returns for overused urgency-based tactics unless they are contextually embedded with realism.

2) Generalized Linear Model Results: To reflect the binary nature of phishing outcomes at the individual level, a binomial logistic regression model was estimated, modelling the probability of compromise $p_i = \Pr(Y_i = 1)$ via the logit link function $\text{logit}(p_i) = \eta_i$, cf. equation (2).

The GLM results confirm the OLS findings while strengthening their inferential validity. *Liking* remains the strongest predictor on the log-odds scale ($\beta_2 \approx 0.92$, $p < 0.001$), followed by *Authority* ($\beta_4 \approx 0.48$, $p = 0.011$). *Scarcity* remains non-significant despite its high prevalence ($p \approx 0.18$), supporting the interpretation that urgency cues may increasingly be ignored due to user desensitization. *Reciprocity* again shows a negative effect ($\beta_1 \approx -0.21$, $p < 0.05$), indicating growing skepticism toward unsolicited assistance.

Overall model performance improves substantially relative to the OLS baseline (McFadden's pseudo-$R^2 \approx 0.19$), indicating that the GLM more accurately captures the behavioural determinants of phishing susceptibility.

3) Interaction Effects Between Influence Principles: Beyond individual effects, additional analyses examined whether combinations of persuasion principles amplify compromise likelihood beyond their isolated contributions. Interaction

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE II
SUMMARY OF INFLUENCE PRINCIPLE PREVALENCE AND STATISTICAL EFFECTS.
ORDINARY LEAST SQUARES (OLS) ON Y_i WITH HC3 ROBUST STANDARD ERRORS ($N = 300$, ADJUSTED $R^2 = 0.126$).
*STATISTICALLY SIGNIFICANT AT $p < 0.05$; **HIGHLY SIGNIFICANT AT $p < 0.001$.

Cialdini Principle	Median Intensity	Prevalence (%)	Regression Coefficient β_k	p-value
Reciprocity	0	11.3%	−0.0263	0.0234*
Liking	1	34.0%	0.6030	<0.001**
Social Proof	0	9.7%	0.0091	0.210
Authority	4	63.7%	0.2011	0.018*
Scarcity	5	72.1%	0.0118	0.081
Commitment/Consistency	3	58.0%	0.0142	0.092

Dependent variable: compromise outcome Y_i; Predictors: persuasion principle intensities X_{ik}; Coefficients β_k represent the marginal effect of principle k on Y_i, holding all other principles constant.

terms of the form $X_{ik}X_{ij}$ were therefore added to the linear predictor η_i, cf. equation (3).

Two interaction effects were statistically significant. The interaction *Authority* × *Scarcity* ($\beta_{45} \approx 0.31$, $p < 0.01$) indicates that hierarchical cues combined with time pressure produce a synergistic increase in compromise likelihood, particularly in BEC and whaling scenarios. The interaction *Liking* × *Social Proof* ($\beta_{23} \approx 0.27$, $p < 0.05$) shows that affective familiarity cues become more persuasive when reinforced by signals that peers or colleagues have already complied, especially in vishing and AI-generated phishing.

No significant interaction effects involving *Reciprocity* were observed, suggesting that this principle operates largely orthogonally to other persuasive cues or may even conflict with dominant phishing strategies.

Interaction coefficients β_{kj} quantify amplification effects beyond the main effects β_k and should therefore not be directly compared in magnitude. Although the standalone effect of *Liking* exceeds the *Liking* × *Social Proof* interaction coefficient, the latter captures a distinct synergistic mechanism. Including interaction terms increases the adjusted R^2 to approximately 0.19, representing a 50% improvement over the OLS baseline.

4) Model Robustness and Comparative Performance: A formal comparison of the three modelling approaches (OLS, GLM, and OLS with interaction terms) was conducted using Akaike's Information Criterion and the Bayesian Information Criterion. The GLM achieves the lowest AIC and BIC values, indicating the most efficient balance between goodness-of-fit and model complexity. For the OLS and GLM models, the number of estimated parameters was $a = 7$ (6 predictors and an intercept). The interaction-augmented model included $a = 22$ parameters, comprising 6 main effects, 15 pairwise interaction terms, and an intercept.

While the interaction-augmented model improves fit relative to the baseline OLS regression, the increased parameter count leads to stronger information-criterion penalties, making it less parsimonious overall. The baseline OLS model performs worst under both criteria.

Robustness checks further confirm the stability of the findings. VIF values remain below 1.6 across all models, HC3 standard errors are stable, and a nonparametric bootstrap (10,000 replications) confirms coefficient stability for the *Liking* and *Authority* predictors.

D. Cluster Analysis of Influence Profiles Across Attack Types

A hierarchical agglomerative clustering (HAC) analysis was conducted using the 10×6 median principle-intensity matrix (cf. Table I) to explore latent structures in how Cialdini's six classical principles manifest across phishing attack types. Ward linkage and Euclidean distances were applied.

1) Cluster A: High-Authority, High-Consistency Attacks: This cluster comprises BEC, CEO-fraud, whaling, vishing, and AI-based spear-phishing, characterized by impersonation of hierarchical roles, sequential escalation of commitments, and the highest observed compromise probabilities.

2) Cluster B: High-Scarcity, Low-Liking Attacks: This group includes generic phishing, smishing, and quishing, characterized by short-message constraints, strong reliance on urgency cues, and limited personalization.

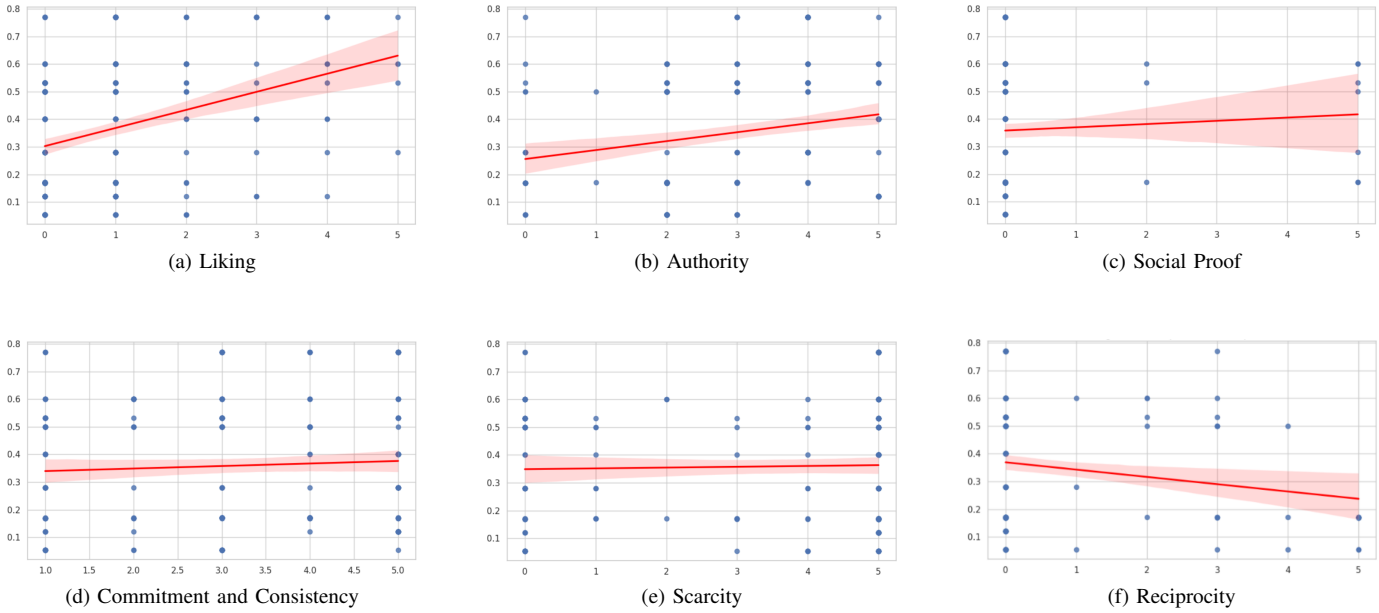
3) Cluster C: Hybrid Personalization Attacks: This cluster consists solely of non-BEC spear-phishing, blending moderate authority cues, moderate liking cues, and contextual personalization.

The clustering results reveal structural attacker strategies not visible in raw principle frequencies and support the interpretation that phishing persuasion is modality-dependent: attacks with constrained communication channels rely disproportionately on scarcity, whereas high-bandwidth channels enable more complex combinations of authority, consistency, and personalization.

E. Dominant Principles and Interactions

Figure 1 presents six linear regression plots, each modeling the relationship between the intensity of a specific Cialdini principle and the corresponding phishing compromise rate. The results reveal substantial variation in behavioral effectiveness. *Liking* demonstrates the strongest positive correlation: higher affective or personalized cues are associated with increased compromise rates, supporting the regression model's identification of liking as the most effective principle. *Authority* also shows a positive linear trend, consistent with its significant regression coefficient, confirming that hierarchical impersonation and institutional tone enhance persuasive success. In contrast, *scarcity*, despite being the most frequently used principle in the dataset, exhibits no meaningful trend, suggesting user desensitization to urgency-based manipulations. *Reciprocity* even displays a negative correlation, pos-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org



(a) Liking (b) Authority (c) Social Proof

(d) Commitment and Consistency (e) Scarcity (f) Reciprocity

Figure 1. Linear regression plots of Cialdini principle intensity (X_{ik}) versus compromise rate (Y_i). While *Liking* and *Authority* exhibit positive associations, *Scarcity* and *Reciprocity* show weak or negative effects. Red shaded areas denote 95% confidence intervals.

sibly reflecting increased user skepticism toward unsolicited favors. *Commitment/Consistency* and *Social Proof* show flat to weakly positive trends, indicating limited predictive utility in isolated message contexts. These results emphasize that influence principle effectiveness is not uniform and may depend on contextual deployment, multimodal layering, or user expectations. Most notably, the data confirm that frequent use does not guarantee behavioral impact.

The statistical findings reveal a decoupling between principle frequency and behavioral effectiveness. Table II shows that *scarcity*, while the most frequently applied principle across all phishing types (median intensity = 5 in 9 out of 10 attack types), did not significantly predict compromise success ($p = 0.081$). In contrast, *liking*, applied in only 34% of the messages, exhibited the strongest statistical association with compromise likelihood ($\beta_2 = 0.6030$, $p < 0.001$).

Similarly, *authority* emerged as both prevalent (63.7%) and significantly effective ($\beta_4 = 0.2011$, $p = 0.018$), particularly in BEC, CEO-Fraud, Whaling, and spear-phishing scenarios where hierarchical power is invoked. Conversely, *reciprocity* (11.3%) showed a negative association with compromise, possibly due to heightened user suspicion of unsolicited "favors" or assistance.

Overall, these findings highlight the importance of focusing not only on principle prevalence but also on behavioral potency and contextual deployment. These findings suggest that the effectiveness of influence principles is highly context-dependent. For instance, while *scarcity* was applied in over 70% of the messages, it showed no statistically significant effect on user compromise. This may be explained by user desensitization in environments where deadline-driven messages are frequent (e.g., corporate inboxes or customer service workflows), reducing perceived urgency. Conversely, *liking*, though less frequent, was particularly effective in personalized or peer-based attacks such as vishing, where social cues are more prominent.

Further, principle efficacy may vary by communication channel: *scarcity* cues are more impactful in constrained formats (e.g., SMS), while *liking* requires richer context or sender familiarity, often found in email or voice interactions. Cultural and organizational factors (e.g., power distance, communication formalism) may also moderate response patterns.

F. Cross-Attack Type Comparison

Comparative analysis across attack types revealed the strategic flexibility of attackers in selecting and combining influence principles. Each phishing vector favors a different psychological profile based on channels, user expectations, and social context.

In *Generic Phishing*, *Scarcity* and *authority* dominate through fake account alerts, service disruptions, and impersonated institutions. Messages are brief and rely on fear and urgency. *Spear-Phishing* exhibits high variance in principle application, often combining *liking*, *authority*, and *commitment*. Personalization is derived from OSINT and contextual familiarity. *BEC* relies on formal tone and impersonation of executives to trigger *authority* and *consistency*. Often embedded in regular workflows (e.g., invoice approvals). In *CEO-Fraud*, top-level executives demand confidential action (e.g., fund transfers). Strongly driven by *authority*, hierarchical obedience, and urgency. *Whaling* targets high-ranking individuals (e.g., C-level execs). Incorporates high contextual detail and cues of exclusivity, invoking *authority*, *scarcity*, and *social proof* (e.g., "Board-only access"). *Clone-/Dynamite-Phishing* uses previously legitimate email threads, duplicated

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

with altered links or attachments. Exploits *commitment* and *consistency* by leveraging past trusted interactions. *Vishing* uses real-time voice to exert psychological pressure. Commonly invokes *authority* (e.g., fake IT or bank staff) and *commitment* by escalating task sequences in live interaction. *Quishing* relies on *commitment* and *scarcity* via QR codes embedded in emails or posters. Users are lured into taking quick action with limited time or context to reflect. *Smishing* emphasizes *scarcity* and *urgency* with short, time-pressured messages. Lacks personalization but achieves reach and immediacy. *AI-Based Phishing* enhances *liking*, *social proof*, and mimics human tone more convincingly, posing new detection challenges.

These patterns illustrate that influence principle deployment is not uniform but highly context-sensitive. For example, *scarcity* dominates in low-bandwidth channels like SMS, whereas *liking* and *commitment* are more effective in richer, interaction-heavy environments such as spear-phishing and vishing. The adaptability of persuasion strategies across attack types reinforces the need for context-aware, psychologically informed defense mechanisms in training, detection, and interface design.

VI. Implications for Cybersecurity Practice

A. Psychological-Aware Security Training

Traditional security awareness programs often focus on technical indicators (e.g., suspicious URLs or attachments), overlooking the psychological mechanisms that drive compliance. The findings demonstrate that *liking* and *authority* are not only prevalent but significantly associated with user compromise. These principles often bypass verification by appealing to trust, familiarity, or hierarchical obedience. Security training must therefore integrate behavioral countermeasures that explain how persuasion operates. For example, users should be taught to question affective signals such as informal greetings from "known" senders or praise followed by requests. Role-playing simulations that mimic real phishing attempts using these principles can foster resistance through experiential learning. Training should also differentiate between the attack types. In high-risk environments (e.g., finance, defense), training must include social engineering reconnaissance awareness and contextual manipulation recognition.

B. Technical Countermeasures and AI Detection

While human awareness is essential, scalable defense requires automated recognition of manipulation patterns. AI-based email filters and NLP can be enhanced to detect rhetorical structures associated with influence principles. For instance, classifiers can be trained to recognize language signaling urgency (scarcity), hierarchical tone (authority), or affective cues (liking) using supervised learning on labeled phishing corpora. As mentioned, current phishing detection systems focus on URL blacklists and attachment scanning. The results suggest that integrating linguistic and psychological features could significantly improve detection precision, especially in text-only or highly targeted attacks. Attention-based models (e.g., transformers) may be particularly effective at identifying subtle combinations of influence tactics across message context.

C. Design Recommendations

Security systems should not only detect threats but also guide users in making safer decisions. Based on the findings, several design strategies are proposed. Contextual warnings can alert users to specific persuasive cues, e.g., "This message may simulate authority", to increase awareness. Cognitive interrupts should be used when requests deviate from normal workflows, such as financial approvals from executives, prompting users to verify intent. Email clients could also highlight rhetorical patterns associated with Cialdini's six classical principles, helping users reassess suspicious messages. For low-bandwidth channels like SMS and QR-based phishing, lightweight NLP can screen for urgency and forcing before users engage.

These interventions represent a move from reactive filtering to proactive behavioral defense, embedding psychological insights into security interfaces to mitigate human-targeted phishing risks created by human cognitive biases.

VII. Conclusion and Future Work

A. Summary of Findings

This study investigated how Cialdini's six principles of persuasion are deployed in phishing attacks and how they translate into behavioural effectiveness. Addressing RQ1, a qualitative analysis of authentic phishing messages revealed that attackers apply persuasion principles strategically rather than randomly. *Authority* is frequently used to simulate hierarchical legitimacy, *liking* to establish interpersonal trust, and *commitment/consistency* to create behavioural momentum. Many messages combine multiple principles, suggesting that psychological synergy plays a central role in successful social engineering.

In response to RQ2, a quantitative content analysis of 300 phishing samples across ten attack types showed that *scarcity* is the most prevalent principle, with high median intensity and presence in more than 70% of the dataset. *Authority* and *commitment/consistency* also appear frequently, particularly in structured fraud scenarios such as BEC, CEO-fraud, and whaling. In contrast, *liking*, *social proof*, and *reciprocity* occur less often overall but are more common in highly personalized attacks, including vishing and AI-generated phishing.

For RQ3, regression-based modelling provided insight into behavioural effectiveness beyond descriptive prevalence. Both the OLS baseline and the binomial logistic regression indicate that *liking* and *authority* are the most robust predictors of compromise likelihood. Conversely, *scarcity*, despite its dominant presence, does not consistently predict compromise, suggesting possible user desensitization to urgency-based cues. *Reciprocity* exhibits a small but statistically significant negative association, which may reflect increasing skepticism toward unsolicited assistance. Interaction analysis further demonstrates

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

that combinations of persuasion principles amplify phishing effectiveness beyond isolated cues, while clustering reveals modality-dependent persuasion profiles across attack types.

Taken together, these findings confirm a central insight of the study: the most frequently deployed persuasion principles are not necessarily the most behaviourally effective. Successful phishing attacks rely on strategically layered affective and hierarchical manipulation rather than urgency alone, underscoring the need for psychologically grounded and context-aware defenses.

B. Methodological Reflection

This work adopts a mixed-method approach that combines qualitative scenario analysis with quantitative statistical modelling. Qualitative inspection of phishing messages enabled contextual interpretation of how persuasion principles are operationalized, while regression analysis, interaction modelling, and unsupervised clustering provided empirical rigor and robustness. The convergence of findings across these methods strengthens confidence in the reported effects.

Nevertheless, several methodological limitations should be acknowledged. First, compromise outcomes were operationalized using empirically assigned rates rather than direct user-level responses, which constrains causal inference. Second, the quantitative analysis relied on publicly available phishing datasets, limiting granularity and potentially introducing selection bias. Third, while pairwise interaction effects were examined, higher-order combinations of persuasion principles were not explored. Despite these constraints, the multi-stage modelling framework improves upon purely frequency-based analyses and offers a more behaviourally informed assessment of phishing effectiveness.

C. Research Extensions

Several directions offer opportunities to extend this work. First, controlled user studies and phishing simulations could be employed to validate the causal impact of individual and combined persuasion principles under realistic conditions. Second, future research should examine how persuasion strategies operate across multimodal attack channels, including email, voice, messaging, and QR-based interactions, as attackers increasingly integrate multiple vectors.

Third, the growing use of large language models for phishing content generation calls for a systematic analysis of how psychologically persuasive language can be produced and detected at scale. Integrating NLP-based feature extraction with behavioural models may enhance detection capabilities against adaptive social engineering. Finally, cross-cultural and organizational studies are needed to assess how contextual norms and institutional structures moderate susceptibility to different persuasion mechanisms, thereby strengthening the generalizability of behaviourally informed phishing defenses. Fourth, future work should extend the coding scheme to include *Unity* as a seventh principle and examine whether it independently predicts compromise beyond the Cialdini's six classical principles (e.g., liking, social proof, and authority).

In addition, future work could explore hybrid models that combine behavioural persuasion features with technical indicators to improve the prediction and mitigation of phishing attacks.

References

[1] A. Lawall, "Quantifying Persuasion – A Comparative Analysis of Cialdini's Principles in Phishing Attacks," in *The Nineteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE) 2025*, 2025, pp. 117–124.

[2] N. Daswani and M. Elbayadi, *Big breaches: Cybersecurity lessons for everyone*. Springer, 2021.

[3] Keepnet, "Top 70 Phishing Statistics and Trends You Must Know in 2025," 10 2024, [retrieved: July, 2025]. [Online]. Available: https://keepnetlabs.com/blog/top-phishing-statistics-and-trends-you-must-know

[4] R. T. Wright, M. L. Jensen, J. B. Thatcher, M. Dinger, and K. Marett, "Research note—influence Techniques in Phishing Attacks: An Examination of Vulnerability and Resistance," *Information Systems Research*, vol. 25, no. 2, pp. 385–400, 2014.

[5] P. Wang and P. Lutchkus, "Psychological tactics of phishing emails," *Issues in Information Systems*, 2023. [Online]. Available: http://dx.doi.org/10.48009/2_iis_2023_107

[6] O. Zielinska, A. Welk, C. B. Mayhorn, and E. Murphy-Hill, "The Persuasive Phish: Examining the Social Psychological Principles hidden in Phishing Emails," in *Proceedings of the Symposium and Bootcamp on the Science of Security*, 2016, pp. 126–126.

[7] R. Cialdini, "Principles and Techniques of Social Influence," *Advanced Social Psychology*, vol. 256, p. 281, 1995.

[8] ——, "Principles of Persuasion," *Arizona State University, eBrand Media Publication*, 2001.

[9] R. Cialdini and B. Sagarin, "Principles of interpersonal influence," *Persuasion: Psychological Insights and Perspectives*, pp. 143–169, 01 2005.

[10] J. Wang, T. Herath, R. Chen, A. Vishwanath, and H. R. Rao, "Research Article Phishing Susceptibility: An Investigation into the Processing of a Targeted Spear Phishing Email," *IEEE Transactions on Professional Communication*, vol. 55, no. 4, pp. 345–362, 2012.

[11] P.-E. Arduin, "To Click or not to Click? Deciding to Trust or Distrust Phishing Emails," in *International Conference on Decision Support System Technology*. Springer, 2020, pp. 73–85.

[12] A. Bergholz, J. De Beer, S. Glahn, M.-F. Moens, G. Paaß, and S. Strobel, "New Filtering Approaches for Phishing Email," *Journal of Computer Security*, vol. 18, no. 1, pp. 7–35, 2010.

[13] C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," in *Ndss*, vol. 10, 2010, p. 2010.

[14] R. Taib, K. Yu, S. Berkovsky, M. Wiggins, and P. Bayl-Smith, "Social Engineering and Organisational Dependencies in Phishing Attacks," in *IFIP Conference on Human-Computer Interaction*. Springer, 2019, pp. 564–584.

[15] R. Ahmad, S. Terzis, and K. Renaud, "Getting users to click: a content analysis of phishers' tactics and techniques in mobile instant messaging phishing," *Information & Computer Security*, vol. 32, no. 4, pp. 420–435, 2024.

[16] K. Uehara, H. Nishikawa, T. Yamamoto, K. Kawauchi, and M. Nishigaki, "Analysis of the relationship between psychological manipulation techniques and both personality factors and behavioral characteristics in targeted email," in *International Conference on Advanced Information Networking and Applications*. Springer, 2020, pp. 1278–1290.

[17] M. De Bona and F. Paci, "A real world study on employees' susceptibility to phishing attacks," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020, pp. 1–10.

[18] M. Butavicius, K. Parsons, M. Pattinson, and A. McCormac, "Breaching the Human Firewall: Social engineering in Phishing and Spear-Phishing Emails," *arXiv preprint arXiv:1606.00887*, 2016.

[19] C. A. Tian, M. L. Jensen, and A. Durcikova, "Phishing susceptibility across industries: The differential impact of influence techniques," *Computers & Security*, vol. 135, p. 103487, 2023.

[20] G. Raywood-Burke, D. M. Jones, and P. L. Morgan, "Maladaptive Behaviour in Phishing Susceptibility: How Email Context Influences the Impact of Persuasion Techniques," in *Human Factors in Cybersecurity*, vol. 91. AHFE International, 2023, pp. 40–49.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[21] D. Oliveira, H. Rocha, H. Yang, D. Ellis, S. Dommaraju, M. Muradoglu, D. Weir, A. Soliman, T. Lin, and N. Ebner, "Dissecting spear phishing emails for older vs young adults: On the interplay of weapons of influence and life domains in predicting susceptibility to phishing," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 6412–6424.

[22] P. A. Lawson, A. D. Crowson, and C. B. Mayhorn, "Baiting the hook: Exploring the interaction of personality and persuasion tactics in email phishing attacks," in *Congress of the International Ergonomics Association*. Springer, 2018, pp. 401–406.

[23] J. Hazell, "Spear Phishing With Large Language Models," *arXiv preprint arXiv:2305.06972*, 2023.

[24] S. C. Matz, J. D. Teeny, S. S. Vaid, H. Peters, G. M. Harari, and M. Cerf, "The Potential of Generative AI for Personalized Persuasion at Scale," *Scientific Reports*, vol. 14, no. 1, p. 4692, 2024.

[25] F. Sharevski and P. Jachim, "Alexa, What'sa Phishing Email?: Training users to spot phishing emails using a voice assistant," *EURASIP Journal on Information Security*, vol. 2022, no. 1, p. 7, 2022.

[26] C. Happ, A. Melzer, and G. Steffgen, "Trick with Treat–Reciprocity increases the Willingness to communicate Personal Data," *Computers in Human Behavior*, vol. 61, pp. 372–377, 2016.

[27] O. Goga, G. Venkatadri, and K. P. Gummadi, "The Doppelgänger Bot Attack: Exploring Identity Impersonation in Online Social Networks," in *Proceedings of the 2015 Internet Measurement Conference*, 2015, pp. 141–153.

[28] A. Vishwanath, T. Herath, R. Chen, J. Wang, and H. R. Rao, "Why do people get phished? Testing individual differences in phishing vulnerability within an integrated, information processing model," *Decision Support Systems*, vol. 51, no. 3, pp. 576–586, 2011.

[29] J. L. Freedman and S. C. Fraser, "Compliance without Pressure: The Foot-in-the-Door Technique," *Journal of Personality and Social Psychology*, vol. 4, no. 2, p. 195, 1966.

[30] H. Abroshan, J. Devos, G. Poels, and E. Laermans, "Phishing Happens Beyond Technology: The Effects of Human Behaviors and Demographics on Each Step of a Phishing Process," *IEEE Access*, vol. 9, pp. 44 928–44 949, 2021.

[31] R. Cialdini, *Influence, New and Expanded: The Psychology of Persuasion*. New York: Harper Business, 2021.

[32] A. Ferreira, L. Coventry, and G. Lenzini, "Principles of Persuasion in Social Engineering and Their Use in Phishing," in *International Conference on Human Aspects of Information Security, Privacy, and Trust*. Springer, 2015, pp. 36–47.

[33] A. Lawall and P. Beenken, "A Threat-Led Approach to Mitigating Ransomware Attacks: Insights from a Comprehensive Analysis of the Ransomware Ecosystem," in *Proceedings of the 2024 European Interdisciplinary Cybersecurity Conference*, ser. EICC '24, S. Li, K. Coopamootoo, and M. Sirivianos, Eds. New York, NY, USA: Association for Computing Machinery, 2024, pp. 210–216. [Online]. Available: https://doi.org/10.1145/3655693.3661321

[34] S. Chanti and T. Chithralekha, "A Literature Review on Classification of Phishing Attacks," *International Journal of Advanced Technology and Engineering Exploration*, vol. 9, no. 89, pp. 446–476, 2022.

[35] A. Lawall, "Fingerprinting and Tracing Shadows: The Development and Impact of Browser Fingerprinting on Digital Privacy," in *The Eighteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE) 2024*, 2024, pp. 132–140.

[36] A. Sumner and X. Yuan, "Mitigating Phishing Attacks: An Overview," in *Proceedings of the 2019 ACM Southeast Conference*, 2019, pp. 72–77.

[37] T. Nisha, D. Bakari, and C. Shukla, "Business E-mail Compromise — Techniques and Countermeasures," in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 2021, pp. 217–222.

[38] M. J. Conyon and L. He, "Executive Compensation and Corporate Fraud in China," *Journal of Business Ethics*, vol. 134, no. 4, pp. 669–691, 2016.

[39] D. Pienta, J. B. Thatcher, and A. Johnston, "Protecting a Whale in a Sea of Phish," *Journal of Information Technology*, vol. 35, no. 3, pp. 214–231, 2020.

[40] S. I. Hashmi, N. George, E. Saqib, F. Ali, N. Siddique, S. Kashif, S. Ali, N. U. H. Bajwa, and M. Javed, "Training Users to Recognize Persuasion Techniques in Vishing Calls," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–8.

[41] T. Vidas, E. Owusu, S. Wang, C. Zeng, L. F. Cranor, and N. Christin, "QRishing: The Susceptibility of Smartphone Users to QR Code Phishing Attacks," in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 52–69.

[42] M. L. Rahman, D. Timko, H. Wali, and A. Neupane, "Users Really Do Respond To Smishing," in *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy*, 2023, pp. 49–60.

[43] J. Jeong, J. Mihelcic, G. Oliver, and C. Rudolph, "Towards an Improved Understanding of Human Factors in Cybersecurity," in *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2019, pp. 338–345.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Secure Software Brownfield Engineering

Aspen Olmsted
School of Computer Science and Data Science
Wentworth Institute of Technology
Boston, MA 02115
email: olmsteda@wit.edu

***Abstract*— The process of securing existing "brownfield" software systems becomes challenging when trying to identify and mitigate vulnerabilities in complex and often undocumented codebases. The paper investigates the essential requirement for improved understanding of program execution flow in legacy PHP applications to support secure software development. The proposed solution uses the PHP extension's trace functionality to dynamically obtain detailed execution paths. The methodology generates complete UML Diagrams by automatically processing program execution trace logs. These diagrams present object-function interactions in visual representations that developers and security analysts use as essential tools. The diagrams provide a straightforward, high-level view of runtime operations, enhancing code understanding and revealing hidden dependencies and security-critical control paths. The automated visualization system helps security professionals detect potential attack vectors, verify the implementation of security controls, and identify insecure data handling practices. The research demonstrates how a debugging tool can be leveraged as a security enhancement tool for brownfield environments, enabling developers to identify vulnerabilities more efficiently without relying on manual code reviews or architectural documentation. This method offers a practical solution to enhance the security posture of legacy applications.**

Keywords- cyber-security; software engineering; secure software development

I. INTRODUCTION

Despite significant advances in secure software engineering, there remains a critical research gap and a strong motivation for practical, low-overhead methods to thoroughly understand and secure legacy ("brownfield") PHP applications. Existing tools and methodologies often fall short when applied to these complex, poorly documented systems, especially in providing actionable insight into runtime behavior and in exposing vulnerabilities. The lack of up-to-date architectural documentation and the evolving nature of these systems create barriers for security analysts and developers alike.

The scientific contribution of this paper is to address the above gap by introducing a novel approach that repurposes XDebug, a widely used PHP debugging tool, to automatically generate various UML diagrams from execution traces. By transforming low-level trace data into high-level visual models, our method enables in-depth analysis of code structure, control flow, and data movement—directly supporting vulnerability detection and security auditing in brownfield environments. Unlike existing static or dynamic analysis tools, our solution is designed for integration into ongoing development and maintenance workflows, requiring minimal overhead and leveraging familiar tools.

The central problem addressed is the lack of effective techniques for reverse-engineering and visualizing the runtime behavior of legacy PHP applications to support security analysis. Specifically, we target the shortcomings of current methods for accurately revealing control pathways, data flows, and architectural drift in systems with incomplete or outdated documentation. Our approach narrows the focus to automating UML model generation from XDebug traces, enabling targeted security assessments and facilitating remediation efforts.

In software development, there is a distinction between developing greenfield systems and maintaining and advancing existing "brownfield " applications. When it comes to greenfield projects, there's the advantage of integrating up-to-date security measures from the start. However, brownfield systems, which comprise the majority of deployed software, pose a significant challenge. These older applications, developed over years or even decades, often lack security protocols, have inadequate documentation, and carry a burden of technical debt. Businesses depend on them for their operations, but their nature and lack of clarity make them vulnerable to security risks. Managing and addressing these vulnerabilities in established settings is challenging because it requires strategies that combine traditional practices with modern security needs.

The primary challenge in securing software lies in its inherently black-box nature. A term used to describe its complex and opaque structure that is difficult to comprehend from the outside perspective alone. Developers and security analysts tasked with securing these systems often face obstacles due to a lack of up-to-date information about how the software operates. The original developers may have moved on to other projects or roles, and the design documents may have changed as well. Even when the codebase is missing altogether, its sheer size can make it overwhelming to navigate efficiently. The system's lack of transparency creates significant difficulties for users in detecting control pathways within the codebase and tracking data movement across software components. The system's lack of transparency creates problems for both identifying vulnerabilities that allow harmful input to enter the system and detecting accidental mishandling of sensitive information.

Traditional analysis tools may highlight problems but often generate numerous false positives or struggle to understand the intricate context of older codebases. On the one hand, manual code inspections are comprehensive. It can be too time-consuming and costly for established applications. Dynamic analysis—observing how a system behaves during operation—provides insight into real-time attributes. However, it often lacks a deep understanding of function calls and object interactions, which is necessary to accurately pinpoint vulnerabilities.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

This article explores the world of long-running PHP applications that have undergone numerous changes and updates over time, often without prioritizing security from the outset. Because of PHP's flexibility and popularity online, these applications have often lacked a security-first approach. An immediate need arises for widely applicable strategies to enhance the security posture of these yet-vulnerable programs.

In our study, we propose a practical method to enhance software development in existing PHP environments by automatically generating multiple UML Diagrams from Xdebug trace logs. XDebug is an open-source tool used by developers to debug and profile PHP code. The robust tracing capability of XDebug primarily serves for debugging and performance evaluation, but it also provides an opportunity for security assessment. XDebug enables running-time information about function calls, method invocations, and variable assignments, providing visibility into program execution behavior.

Our goal is to repurpose this well-known developer tool to help you understand complex legacy code with minimal overhead and maximum effectiveness. This work is an extension of a previously published conference paper [1].

Our approach includes a series of steps to achieve our goal efficiently. First, we enable XDebug tracing for scenarios or attacks in the existing PHP system under consideration, creating logs of the exact sequence of actions taken. Secondly, we programmatically. Process these logs to extract essential details, such as function names, class methods, arguments, return values, and execution order. We transform the extracted data into a format that works for creating UML Sequence Diagrams. The visual representations illustrate how individual objects and functions interact over time, demonstrating the control flow and data movement within the software application.

There are many benefits to using reverse-engineered UML Sequence Diagrams in software engineering. To begin with, the advantages they bring forth. These diagrams help enhance understanding of code structures. This is especially valuable for developers or security experts who must decipher intricate legacy codebases, as the sequence diagram provides a holistic overview of how components collaborate to accomplish specific functionalities. This visual representation simplifies comprehension compared to sifting through lines of undocumented code. Furthermore, diagrams reveal hidden dependencies and unforeseen interactions within the system. In systems that have undergone development or modification (brownfield systems), functions may interact in unexpected ways, or data may pass through intermediate steps or components that are not immediately apparent. Sequence diagrams bring clarity to these hidden connections by spelling out the relationships, for a deeper examination of possible repercussions or unintentional data disclosures.

From a security standpoint, these diagrams highlight control pathways. By showing the sequence of actions, analysts can easily identify areas of attack, such as points where user input is handled, where external data is used, or where essential tasks are performed. They can assist in tracking how unreliable data moves, from its entry point to processing steps, exposing points for injection or flaws in deserialization security. Additionally, diagrams help confirm that security measures are properly implemented. For example, a person can visually verify whether authentication checks are performed in real time, whether input validation is used regularly, or whether authorization decisions are made before accessing resources. This automated display enables security teams to conduct efficient security evaluations, reducing the need for tedious manual techniques.

The primary objective of this study is to demonstrate that repurposing XDebug for automated UML model generation provides a practical, effective means for security analysis of legacy PHP applications. By converting runtime execution data into high-level visualizations, we equip security professionals and developers with actionable insight into complex systems, facilitating targeted vulnerability discovery and risk mitigation. This approach reduces reliance on manual documentation, supports the maintenance and evolution of brownfield codebases, and enhances security.

To guide the reader, the remainder of the paper is structured as follows. Section II reviews related work and the limitations of current methods. Section III presents a motivating example. Section IV details the implementation of our parser. Section V describes the generation of sequence diagrams. Section VI covers communication diagrams for application coupling. Section VII discusses activity diagrams to illustrate business-logic workflows. Section VIII addresses component diagrams for managing architectural drift. Section IX explores timing diagrams to visualize performance-related security risks. Section X analyzes the use of execution trace fields. Section XI concludes and outlines directions for future work.

II. RELATED WORK

Secure software engineering has made substantial progress through greenfield development, which enables security integration at the beginning of software development. The distinctive obstacles of brownfield systems require separate attention. This section examines relevant literature on secure software development, with particular attention to research on integrating security into existing codebases and applying dynamic analysis and visualization techniques.

A foundational aspect of secure software engineering is the proactive integration of security considerations throughout the software development lifecycle. Our seminal work, "Security Driven Software Development" [2], provides a comprehensive framework for embedding security into every phase of development, from requirements gathering to deployment and maintenance. This book emphasizes the importance of a security-first mindset and offers strategies for identifying and mitigating risks early. While primarily focused on new development, the principles outlined by Olmsted, such as threat modeling and secure coding practices, are equally relevant to brownfield remediation efforts. Our proposed methodology, which aims to improve understanding of existing brownfield code, directly supports the application of such security principles by making the implicit explicit.

The paper by Olmsted titled "Secure software development through non-functional requirements modeling" [3] expands on the significance of early security integration by demonstrating that non-functional requirements (NFRs) are essential to

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

software security. The paper indicates that security requirements should be treated as NFRs and explicitly modeled during the initial development phase.

The precision and verifiability of security requirements can be enhanced through the use of formal specification languages such as Object Constraint Language (OCL) and UML stereotypes in this context [2]. For brownfield systems, where NFRs may not have been formally captured during initial development, our approach to generating UML Sequence Diagrams helps reverse-engineer the system's behavior. By visualizing execution flows, it is possible to infer how security-related NFRs (e.g., access control, input validation) are currently handled or where they are conspicuously absent. The analysis results will guide the redefinition of security NFRs and direct remediation.

The paper "Secure Software Development–Models, Tools, Architectures and Algorithms" by Olmsted [4] presents a comprehensive overview of the elements necessary to develop secure software systems. This paper discusses multiple security models, the tools that aid analysis, architectural considerations for building secure systems, and the algorithms underlying security mechanisms. Our research aligns with this broader vision by introducing a practical tool-based approach (leveraging XDebug) to generate a specific model (UML Sequence Diagrams) that aids in understanding the architecture and behavior of brownfield PHP applications. The generated diagrams serve as essential inputs for security models and algorithm applications, helping analysts detect hidden vulnerabilities in complex legacy code by tracing data and control flow.

While existing literature extensively covers secure software development, a gap remains in practical, low-overhead methods for thoroughly understanding the runtime behavior of brownfield applications for security purposes. Static analysis tools (e.g., SAST solutions) are effective at identifying patterns of vulnerabilities but often struggle with context and produce false positives in complex legacy code. The interaction of Dynamic Application Security Testing (DAST) tools with running applications reveals vulnerabilities, but it operates at a higher abstraction level than XDebug traces provide function-level details. Our solution enhances existing tools by creating detailed visual execution flow maps that aid manual security auditing, threat modeling, and vulnerability impact assessment for complex brownfield PHP applications. Repurposing XDebug for this task provides a unique advantage by leveraging a widely available, familiar developer tool, reducing the learning curve and integration overhead for teams working with legacy PHP systems.

Functional requirements can be defined and represented in various ways. While these requirements serve as the foundation for software development, non-functional requirements (NFRs) provide the essential guidelines for coding implementation. Many authors have examined NFRs and the challenges of incorporating them into the design process. Pavlovski and Zou [5] NFRs are defined as specific behaviors and operational constraints, including performance expectations and policy limitations. Despite many discussions surrounding them, they are often not given the attention they deserve. In our work, these NFRs can be tagged with UML stereotypes.

Glinz [6] suggests categorizing functional and non-functional requirements to ensure they are considered in application development. Alexander [7] points out that the language used to describe requirements is essential, noting that words ending in "-ility," such as reliability and verifiability, often refer to NFRs. Much of this research focuses on identifying NFRs. Our work builds on these foundations by providing applied tools for brownfield engineers to tag NFR requirements in the existing codebase.

Ranabahu and Sheth [8] explore four different modeling semantics to represent cloud application requirements: data, functional, non-functional, and system. Their work primarily addresses functional and system requirements, with some overlap in non-functional requirements from a system perspective. They built upon research conducted by Stuart, who defined semantic modeling languages for modeling cloud computing requirements throughout the three phases of the cloud application life cycle: development, deployment, and management. Our work fills in the gap in implementation of non-functional requirements.

Ranabahu and Sheth [8] use Unified Modeling Language (UML) to model only functional requirements. UML [9] is a standardized notation for representing the interactions, structures, and processes of software systems. It consists of various diagram types, with individual diagrams linked to different perspectives of the same part of a software system. We utilize UML to visualize the existing brownfield codebase..

The Object Constraint Language (OCL) [10] is part of the official Object Management Group (OMG) standard for UML. An OCL constraint specifies restrictions for the semantics of a UML specification and is considered valid as long as the data is consistent. Each OCL constraint is a declarative statement in the design model that signifies correctness. The constraint is expressed at the class level, while enforcement occurs at the object level. Although OCL has operations to observe the system state, it does not include functions to modify it. After our generation of the UML models, OCL can be applied to define correctness in the current codebase.

III. MOTIVATING EXAMPLE

Our proposed methodology demonstrates practical utility through an example that focuses on SuiteCRM, an open-source Customer Relationship Management (CRM) system widely used by many organizations. The open-source PHP application SuiteCRM is an excellent brownfield example, with extensive complexity from multiple years of development and incomplete modern architectural documentation. The complexity of SuiteCRM's functionality makes it challenging for new developers and security auditors to understand its security aspects. Our method for creating UML Sequence Diagrams from XDebug traces enables effective business process reverse-engineering, improving code understanding and security analysis.

The following common user scenarios in SuiteCRM demonstrate how program traces reveal their execution flows:

1. Scenario 1: User Creates Contact

- User Action: A sales representative navigates to the "Contacts" module, fills in various contact

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

details (e.g., name, email, phone number, address), and submits the form to save the new contact record.

- Trace Insight and Security Relevance: When XDebug tracing is enabled during this operation, the generated trace log meticulously records every function call, method invocation, and file inclusion that occurs from the moment the form submission is processed. This includes the initial handling of the HTTP POST request, validation routines, and, critically, the data persistence logic. The trace would capture calls to files such as, modules/Contacts/Save.php, revealing the sequence of operations involved in taking the submitted data and committing it to the database.
- A detailed analysis of the sequence diagram derived from this trace shows:
- Input Handling: How the raw form data is received and sanitized (or not) before processing. This is crucial for identifying potential Cross-Site Scripting (XSS) or SQL Injection vulnerabilities if input validation is insufficient or bypassed.
- Data Flow: The path of sensitive contact information (e.g., email addresses, personal details) as it moves from the web form, through various PHP functions, and ultimately to the database. This helps in understanding where data might be exposed or mishandled.
- Database Interaction: The specific functions responsible for constructing and executing SQL queries for insertion into the contacts table. Insecure practices, such as direct string concatenation in SQL queries, would be immediately apparent, highlighting SQL injection risks.
- Workflow Triggers: If the creation of a contact triggers other business logic (e.g., sending a welcome email, updating related accounts, or initiating a workflow), the trace would show the invocation of these subsequent functions. This helps understand the full impact of a contact-creation operation and identify any security implications of these cascading actions (e.g., unauthorized email sending).
- Access Control: The diagram could reveal where authorization checks are performed (or omitted) before data is saved, indicating potential Insecure Direct Object Reference (IDOR) or unauthorized data modification vulnerabilities if a user can manipulate data they shouldn't.

2. Scenario 2: User Schedules a Meeting

- User Action: A user accesses the "Meetings" module, enters details such as the meeting subject, time, date, duration, and invites participants (e.g., other users, contacts, leads), then saves the meeting record.
- Trace Insight and Security Relevance: Tracing this scenario would provide a rich sequence of interactions involving the logical meeting module and its dependencies. The trace illustrates how meeting details are processed, how participants are associated, and how notifications may be generated.
- The resulting sequence diagram would be invaluable for:
- Participant Management: Understanding how participants are linked to the meeting. This is critical for assessing potential information leakage (e.g., if a user can view participants, they shouldn't) or unauthorized access to meeting details.
- Cross-Module Interactions: Visualizing the calls to linked modules, such as Users and Contacts, to retrieve participant information. This helps identify potential privilege-escalation paths when the system implicitly trusts data retrieved from these modules without proper revalidation.
- Calendar Integration: If the meeting scheduling integrates with an internal calendar or external service, the trace would expose the functions responsible for these interactions. This enables security analysis of data exchanged with external systems.
- Notification Mechanisms: Tracing the functions responsible for sending meeting invitations or reminders. This can reveal vulnerabilities related to email spoofing, content injection in notifications, or denial-of-service if the notification system is abused.
- Time and Date Handling: How time and date inputs are processed and stored. Incorrect handling of time zones or date formats can lead to logical flaws or even a denial-of-service attack if parsing errors are not handled gracefully.

3. Scenario 3: User Creates an Invoice

- User Action: An accountant generates a new invoice through the "Invoices" module, associating it with a specific client (Account), adding various products or services, specifying quantities and prices, and saving the invoice.
- Trace Insight and Security Relevance: This scenario is particularly sensitive due to its financial implications. The XDebug trace would capture the complex interactions involved in creating invoice entries, calculating totals, and establishing relationships between Accounts, Products, and Invoices modules.
- The sequence diagram would reveal:
- Financial Calculation Logic: The precise functions involved in calculating line item totals, taxes, and the grand total of the invoice. This is paramount for identifying potential manipulation vulnerabilities (e.g., rounding errors, incorrect tax calculations, or

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

unauthorized price modifications) that could lead to financial discrepancies.
- Relationship Management: How the invoice is linked to an Account (client) and Products. This helps in understanding access control mechanisms for financial data and preventing unauthorized association of invoices with incorrect clients or products.
- Data Integrity: Tracing the flow of product quantities, prices, and client details into the invoice. Any points where these values are not adequately validated or where they could be tampered with before persistence would be highlighted.
- State Transitions: If an invoice goes through different states (e.g., Draft, Pending, Paid), the trace shows the functions responsible for these state changes, allowing for analysis of potential unauthorized state transitions.
- Reporting and Export: If invoice creation triggers the generation of a PDF or an export to an accounting system, the trace would expose the functions handling this, allowing for security review of data serialization and external communication.

In each scenario, the automatically generated UML Diagrams provide a visual roadmap of the application's runtime behavior. This "living documentation" is far more accurate and up to date than static, manually created diagrams, which often become obsolete as the codebase evolves. For brownfield applications like SuiteCRM, these diagrams transform opaque execution paths into transparent, analyzable flows, significantly reducing the time and effort required for security auditing, vulnerability discovery, and targeted remediation. They empower security professionals and developers to ask precise questions about data handling, access control, and business logic, ultimately leading to a more secure and resilient system.

IV. PARSER IMPLEMENTATION

The core of our methodology lies in the ability to accurately parse and interpret the detailed trace logs generated by XDebug. This section describes the implementation of our parser, which transforms raw, verbose XDebug output into a structured, actionable format suitable for generating UML Sequence Diagrams.

XDebug trace files, typically in the .xt format, contain a chronological record of every function call, method invocation, file inclusion, and variable assignment during a PHP script's execution. While incredibly rich in detail, their raw format is not directly consumable by UML diagramming tools. Our Java parser addresses this by extracting salient information and organizing it into a programmatic representation that captures the essential elements of a sequence diagram: lifelines (objects/functions), messages (method calls), and their temporal order.

1. XDebug Trace File Format Overview

Before detailing the parser's design, it's essential to understand the structure of XDebug trace files. XDebug offers several trace formats, but the most common and detailed is the "computer readable" format (format 1). Each line in this format represents an event (e.g., function entry, function exit, include, require, eval) and contains a series of tab-separated fields. Key fields include:

- Level: The nesting level of the function call.
- Function Number: A unique identifier for the function call instance.
- Type: Indicates the event type (e.g., 0 for function call, 1 for function return, 2 for include, 3 for require, 4 for eval).
- Function Name: The name of the function or method being called.
- File Name: The PHP file where the function call originated.
- Line Number: The line number within the file.
- Time: Timestamp of the event.
- Memory: Memory usage at the time of the event.
- Arguments: A representation of the arguments passed to the function (if configured to be included).

2. Parser Design and Architecture

Our Java parser is designed as a modular component, following a typical parsing pipeline: reading, lexical analysis, syntactic analysis, and data model construction.

3. File Reading and Line-by-Line Processing:

The parser begins by reading the XDebug trace file line by line. Given that trace files can be very large (hundreds of megabytes for complex operations), an efficient line-by-line reading mechanism is crucial to avoid excessive memory consumption. Java's BufferedReader is employed for this purpose.

4. Lexical Analysis (Tokenization):

Each line read from the trace file undergoes lexical analysis. Since the fields are tab-separated, a simple String.split("\t") operation is sufficient to break down each line into its constituent tokens. Robust error handling is implemented to handle malformed lines or unexpected field counts, preventing parser crashes caused by corrupted trace data.

5. Syntactic Analysis and Event Interpretation:

After tokenization, the parser performs syntactic analysis by interpreting each token's meaning based on its position and the event Type field. A switch statement or a strategy pattern can be used to handle different event types (0 for call, 1 for return, etc.).

Function/Method Calls (Type 0): When a function call event is encountered, the parser extracts the function name, the originating file and line number, and the call level. This information is used to identify the "caller" and "callee" in the sequence. The Function Number is critical for matching function calls with their corresponding returns.

Function/Method Returns (Type 1): Upon encountering a function return event, the parser uses the Function Number to locate the corresponding outstanding function call. This pairing is essential for determining a call's duration and correctly nesting messages in the sequence diagram.

Includes/Requires (Type 2, 3): These events indicate file inclusions. While not direct messages in a UML Sequence Diagram, they are important for understanding the context and

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

dependencies within the PHP application. The parser can record these events to provide additional context or to help in identifying the "lifeline" associated with the executed code.

6. Data Model Construction:

The most critical phase is the construction of an in-memory data model that represents the sequence of interactions. We define several Java classes to represent the elements of a UML Sequence Diagram:

7. SequenceDiagram: The top-level class representing the entire diagram, containing a list of lifelines and messages.

Lifeline: Represents an object or function participating in the sequence. For PHP, this typically maps to a class name, an object instance, or a global function. The parser dynamically creates lifelines as new, unique function or method owners are encountered.

Message: Represents communication between two lifelines. Key attributes include:

- sender: The Lifeline initiating the message.
- receiver: The Lifeline receiving the message.
- methodName: The name of the function/method being called.
- callLevel: The nesting depth of the call.
- startTime: Timestamp of the call.
- endTime: Timestamp of the return.
- arguments: (Optional) Parsed arguments.
- returnValue: (Optional) Parsed return value.
- messageType: (e.g., synchronous call, return).

The parser maintains a stack-like structure (e.g., a Deque or Stack in Java) to keep track of current active function calls. When a Type 0 event (call) occurs, a new Message object is created and pushed onto the stack. When a Type 1 event (return) occurs, the corresponding Message is popped, its endTime is set, and it is added to the Sequence Diagram's list of messages. This stack-based approach correctly handles nested function calls and ensures the proper temporal ordering of messages.

V. UML Sequence Diagram Generation

Once the XDebug trace data has been successfully parsed into our structured Java data model (comprising Sequence Diagram, Lifeline, and Message objects), the next step is to translate this model into a visual UML Sequence Diagram. For this purpose, we use PlantUML, a powerful open-source tool that lets users create UML diagrams from a simple, human-readable text description.

The choice of PlantUML offers several significant advantages:

- Text-Based Definition: Diagrams are defined in plain text, making them easy to generate programmatically, version-controlled, and collaboratively worked on. This aligns well with automated generation from trace files, as our Java parser can directly output the PlantUML syntax.
- Ease of Integration: PlantUML can be integrated into various environments and workflows. The generated text file can be rendered as images (PNG or SVG) or in other formats using the PlantUML command-line tool, a dedicated server, or IDE plugins.
- Flexibility and Expressiveness: PlantUML supports a wide range of UML diagram types, including Sequence Diagrams, with rich features for actors, participants, messages, activation bars, loops, conditionals, and notes, allowing for detailed and expressive visualizations.

Our Java parser, after constructing the Sequence Diagram object, includes a component that generates PlantUML syntax. This component iterates over Lifeline and Message objects in the data model and translates them into PlantUML syntax.

- Participants/Lifelines: Each unique Lifeline object identified during parsing (e.g., a class name like ContactService, DatabaseHandler, or a generic Application for global functions) is declared as a participant in PlantUML using keywords like participant, actor, or boundary.
- Messages: Each Message object is translated into a PlantUML message arrow. The sender and receiver lifelines determine the arrow's source and target, and the methodName becomes the message label. Activation bars are automatically handled by PlantUML when -> (call) and <- (return) messages are used.
- Nesting and Call Levels: The callLevel attribute of our Message objects is crucial for correctly representing nested calls and activation bars. PlantUML inherently handles nesting through the sequence of -> and <- messages, but explicit activate and deactivate keywords can be used for finer control.
- Conditional Logic and Loops: While XDebug traces capture the executed path, they don't directly provide information about if conditions or for loops that weren't taken. However, for executed loops or branches, the repeated messages or specific sequences can be grouped using PlantUML's loop or alt/else constructs, which can be inferred from patterns in the trace or added manually for clarity.

The output of this component is a plain text file (e.g., diagram.puml) containing the PlantUML definition. This file can then be fed into a PlantUML renderer to produce the final visual sequence diagram, providing an intuitive and accurate representation of the brownfield application's runtime behavior. This automated generation significantly reduces the manual effort traditionally required to create and maintain such diagrams, making them a practical tool for security analysis and code comprehension.

VI. UML Communication Diagram Generation

While the generated Sequence Diagrams excel at visualizing the temporal order of operations, understanding the structural relationships and coupling between system components is equally critical for securing brownfield applications. To address this, we extend our methodology to generate UML Communication Diagrams (formerly known as Collaboration Diagrams).

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Rationale and Security Value Communication diagrams abstract away the time dimension to focus on the links between objects. In the context of legacy PHP applications, these diagrams are instrumental in identifying "God objects" (classes that know too much or do too much) and visualizing the "attack surface" of specific modules. By aggregating interactions, security analysts can instantly see which components communicate with sensitive data handlers, such as the Database or Auth modules.

Parser Adaptation To generate these diagrams, we utilize the same in-memory data model constructed by our Java parser. However, rather than iterating through the Message list linearly, the parser performs an aggregation step:

- Interaction Aggregation: The parser traverses the list of Message objects, using the sender and receiver attributes to build an adjacency matrix or a graph structure.
- Frequency Analysis: We count the frequency of calls between unique pairs of lifelines. This metric helps identify tight coupling and potential bottlenecks.

PlantUML Generation: The parser translates this graph into PlantUML syntax. Unlike the sequence diagram, the layout here focuses on nodes (objects) and connectors (links).

- Nodes: The Lifeline objects (e.g., ContactService) are rendered as rectangle or agent components.
- Links: A single link is drawn between any two interacting objects. The parser annotates this link with the direction of flow and the list of method names invoked.

VII. UML Activity Diagram Generation

Sequence and Communication diagrams focus on objects and messages. However, security auditing often requires a clear understanding of the business logic workflow—decisions, parallel processing, and the "swimlanes" of responsibility. To achieve this, we further adapt our XDebug trace analysis to generate Dynamic Activity Diagrams.

Handling Linear Trace Limitations - It is important to note that an XDebug trace represents a single, linear execution path—a concrete instance of the program running. Therefore, the Activity Diagram generated does not show every possible branch (e.g., the else condition that was not taken). Instead, it visualizes the actual execution flow of a specific scenario. This "forensic" view is highly valuable for incident response, allowing analysts to see exactly which logic path an attacker's input took.

Parser Implementation for Activity Flows. We map the XDebug trace elements to Activity Diagram components as follows:

- Swimlanes: The File Name or Class Name from the trace determines the swimlane. This visually separates responsibilities, showing control passing from the Controller swimlane to the Service swimlane, and then to the Repository swimlane.
- Activities: Each function call represents an action state.
- Control Flow: The sequence of function calls defines the arrows connecting activities.

PlantUML Syntax Generation - the Java parser iterates through the events and generates PlantUML activity syntax:

- Swimlane Declaration: |ClassName| is used to define the context.
- Action Definition: start, stop, and : Action Name; denote the flow.

Logical Grouping: Although the trace is linear, the parser can infer logical groupings. For example, if the trace enters a specific validation class and executes multiple internal methods before returning, the parser can wrap these methods in a PlantUML partition block labeled "Validation" to improve readability.

VIII. UML Component Diagram Generation

While Sequence and Communication diagrams detail object-level interactions, brownfield systems often suffer from "architectural drift," in which the boundaries between high-level modules degrade over time. To address this, we can abstract the trace data to generate UML Component Diagrams.

Abstraction Strategy - Unlike previous diagrams that map one-to-one with classes or objects, Component Diagrams require grouping strategies. Our parser utilizes the file path and namespace information captured by XDebug to infer component boundaries.

- Directory-Based Grouping: In many legacy PHP frameworks (like SuiteCRM), directory structures often mirror architectural modules (e.g., modules/Contacts, include/database). The parser aggregates all classes found within a specific directory into a single UML [Component].
- Namespace-Based Grouping: For modern PSR-4 compliant code, the parser groups classes by their top-level namespace (e.g., Vendor\Module).

Inter-Component Dependency Mapping - The parser iterates through the trace's call stack. When a function in Component A calls a function in Component B, a dependency link is established.

Security Value: This visualization is critical for "Architecture Validation." It immediately reveals whether the Presentation Layer bypasses the Service Layer to access the Data Layer directly. It helps identify circular dependencies and "spaghetti code" that complicates security patching.

PlantUML Implementation - The parser output simplifies the thousands of function calls into high-level relationships:

- Definition: Components are defined using brackets: component [Contacts Module].
- Relationships: Dependencies are drawn based on aggregated calls: [Contacts Module] ..> [Database Core] : uses.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

IX. UML Timing Diagram Generation

Standard XDebug traces capture precise timestamps for every function entry and exit. While Sequence Diagrams show the order of events, they often obscure the duration. To visualize performance-related security risks, we propose generating UML Timing Diagrams.

Leveraging Timestamp Data. The parser uses the Time and Level fields from the trace logs to calculate the "execution width" of specific critical functions.

- State Changes: The diagram treats the active execution of a function as a "state."
- Duration Visualization: The horizontal axis represents real time (in milliseconds or microseconds). The parser plots the duration of a specific lifeline (e.g., an Authentication Service) that remains in an "Active" state.

Security and Performance Analysis - This diagram type is uniquely suited for identifying availability risks:

- DoS Vulnerabilities: It highlights functions that consume disproportionate execution time, which could be targets for Algorithmic Complexity Denial of Service (ReDoS) or resource exhaustion attacks.
- Timing Attacks: In sensitive cryptographic or authentication routines, the diagram can help visualize variances in execution time based on different inputs, potentially exposing side-channel vulnerabilities.

PlantUML Implementation - The parser generates "robustness" timing diagrams:

- Clock Definition: The parser scales the XDebug timestamps to a relative scale for the diagram.
- State Definitions: robust "AuthService" as WC defines the participant.
- Transitions: Syntax such as @100 <-> @450 : {validating} visualizes exactly how long the validation routine held the thread.

X. Trace File Mapping

To demonstrate the versatility of our parsing methodology, Table 1 illustrates how specific fields from the XDebug "computer readable" format are repurposed to drive the generation of different UML diagram types. While the raw data remains constant, the parser's interpretation of fields—such as File Name and Time—shifts based on the desired visualization output.

Parser Implementation Notes

- Sequence & Timing: Rely heavily on the Time and Level fields to reconstruct the exact runtime "shape" of the execution.
- Component & Activity: Utilize the File Name field not just for identification, but for logical grouping. For Component diagrams, the parser truncates file paths to group classes by directory (e.g., modules/Contacts/), while Activity diagrams use the class name to assign "swimlanes".
- Communication: Ignores the Time field entirely to focus on the frequency of distinct Function Name pairs between File Name entities, aggregating multiple calls into single association links.

XI. Limitations and Threats to Validity

While our approach demonstrates practical value for brownfield security analysis, several limitations and threats to validity should be acknowledged to increase transparency and scientific rigor:

- Dependence on Single Execution Traces: The generated UML diagrams are derived from individual XDebug trace files, each representing a single execution path through the application. As a result, they visualize only the specific scenarios or workflows executed during tracing. Branches of logic, error paths, and rare execution flows not triggered in the captured scenario will not appear in the diagrams, potentially omitting critical vulnerabilities or behaviors.
- Incomplete Coverage: Comprehensive security analysis requires exercising all relevant application features and edge cases. If the trace does not cover certain code paths, those paths will not be reflected in the reverse-engineered diagrams, leading to incomplete architectural or security insights. Achieving exhaustive coverage in large legacy systems is often impractical.
- Dynamic Trace to UML Mapping Inaccuracies: The translation from dynamic trace events to UML diagram elements inevitably involves abstraction and inference. Complex PHP features such as dynamic method calls, anonymous functions, and runtime code evaluation (eval) may not be perfectly captured. Some architectural elements or control/data flows may be misrepresented or oversimplified.
- Environmental and Configuration Dependencies: The accuracy and granularity of the trace data depend on XDebug configuration and PHP runtime settings. Certain features (e.g., argument capturing, extended event types) may be disabled or inconsistent across environments, affecting the completeness and fidelity of generated diagrams.
- Assumptions about Codebase Consistency: The approach assumes the codebase under analysis is in a runnable state and that the observed execution traces are representative of typical operation. Highly dynamic codebases or those with substantial runtime variability may yield traces that are hard to generalize.
- Manual Post-Processing Needs: For some advanced diagram types (e.g., component and timing diagrams), current automation is incomplete, and manual intervention or interpretation may be required to ensure accuracy and usefulness.
- Future work should address these limitations by exploring automated ways to merge multiple traces, improve coverage measurement, and refine the mapping algorithms to better handle complex PHP constructs and architectural patterns.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Table 1 - Trace Field Usage

XDebug Field	Field Description	Sequence Diagram (Temporal Flow)	Communication Diagram (Object Coupling)	Activity Diagram (Business Logic)	Component Diagram (Architecture)	Timing Diagram (Performance/DoS)
Function Name	Name of the called function/method	Message Label (the request being sent)	Link Label (the interaction type)	Action State (the task being performed)	Dependency Indicator (implies usage)	State Label (the active process)
File Name	Originating PHP file path	Lifeline Identity (Class/Object)	Node Identity (Object)	Swimlane (Responsibility Area)	Component Boundary (Directory/Namespace Grouping)	N/A
Time	Timestamp of the event	Ordering of messages (vertical axis)	N/A (Abstracted away)	Implicit ordering of flow	N/A	Duration (Width of state on horizontal axis)
Level	Nesting depth of the call	Activation Bars & Nested Flows	Direction of dependency	Sub-activity grouping	Dependency Direction	State Change (Active vs. Idle)
Type	Event type (Entry, Exit, Include)	Arrow Direction (-> vs <-)	N/A	Control Flow (Start/Stop)	N/A	State Transition Trigger
Memory	Memory usage at event time	Optional Note (Resource consumption)	N/A	N/A	N/A	Resource Plot (Secondary Y-axis analysis)

XII. Conclusions and Future Work

Future work should address multiple challenges, starting with large file sizes, in our research findings. XDebug trace files tend to expand their size when complex operations or scripts run for extended periods. The parser needs both memory efficiency and the ability to handle files that exceed RAM capacity. Some specific parser improvements include:

- The trace data produced by XDebug includes all function arguments in its output. Interpreting complex PHP data structures (arrays, objects) in Java requires advanced logic to convert them into meaningful Java representations. The initial development used basic data types, but it should be expanded to handle logging of complex arguments.
- Performance is vital when handling massive trace files. String manipulation using StringBuilder, along with efficient data structures and minimizing object creation, will substantially boost performance.
- Real-world trace files may contain corrupted lines or unexpected formatting due to system crashes or misconfigurations. The parser needs to maintain robustness by either skipping malformed entries or logging warnings to prevent crashes.
- Converting PHP dynamic elements, such as global functions, anonymous functions, and closures, into standard UML lifelines and messages requires thorough analysis. The system should treat global tasks as part of a basic "Application" lifeline and each object should receive its own lifeline based on its class name.

Our solution transforms XDebug trace data via programmatic parsing, then generates PlantUML diagrams to create structured programmatic models that become visualized. The generated models are valuable visual diagrams for analyzing and securing brownfield applications. The parser architecture enables future extensions of XDebug features while providing flexibility for processing various trace formats. A primary next step in our work is to inject risk-mitigation code into the brownfield codebase. Some examples include:

- Code injection to enforce OCL invariants – This work will ensure invariants always evaluate to true.
- Code injection to enforce OCL pre-conditions - This work will ensure pre-conditions are checked and evaluated to true before messages are processed.
- Code injection to enforce OCL post-conditions- This work will ensure post-conditions are checked and evaluated to true after messages are processed.
- UML stereotype risk mitigation code injection - This work will allow the definition of risk mitigation code for the different places UML stereotypes can be tagged.

References

[1] A. Olmsted, "Secure Software Brownfield Engineering – Sequence Diagram Identification," in *Proceedings of The*

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Nineteenth International Conference on Emerging Security Information, Systems and Technologies, Barcelona, Spain, 2025.

[2] A. Olmsted, Security-Driven Software Development: Learn to analyze and mitigate risks in your software projects, Birmingham, UK: Packt Publishing, 2024.

[3] A. Olmsted, "Secure software development through non-functional requirements modeling," in *Proceedings of the 2010 International Conference on Software Engineering Research and Practice (SERP)*, 2010.

[4] A. Olmsted, "Secure Software Development–Models, Tools, Architectures and Algorithms," *Journal of Software Engineering and Applications,* pp. 743-750, 2012.

[5] C. J. Pavlovski and J. Zou, "Non-functional requirements in business process modeling," *Proceedings of the Fifth on Asia-Pacific Conference on Conceptual Modelling, vol. 79,* pp. 1-10, 2008.

[6] M. Glinz, "Rethinking the Notion of Non-Functional Requirements," *Third World Congress for Software Quality, Munich, Germany,* pp. 1-10, 2005.

[7] I. Alexander, "Misuse Cases Help to Elicit Non-Functional Requirements," *Computing & Control Engineering Journal, 14, 40-45,* pp. 1-10, 2003.

[8] R. Ajith and A. Sheth, "Semantic Modeling for Cloud Computing, Part I," *Computing,* vol. May/June, pp. 81-83, 2010.

[9] Object Management Group, "Unified Modeling Language: Supersturcture," 05 02 2007. [Online]. Available: http://www.omg.org/spec/UML/2.1.1/. [Accessed 20 Feb 2025].

[10] Object Management Group, "OMG Formally Released Versions of OCL," 02 2014. [Online]. Available: http://www.omg.org/spec/OCL/. [Accessed 20 02 2025].

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

User-Centric Privacy in Smart Healthcare: Cross-Service Analysis and Design Implications

Elias Seid, Huan Guo, Fredrik Blix, Yuhong Li
Department of Computer and Systems Sciences
Stockholm University, Sweden
(elias.seid) @dsv.su.se

***Abstract*—Privacy in smart healthcare is not merely a technical problem; it is a perceptual one. Users routinely encounter protection measures they cannot see, understand, or control, and this gap erodes trust and willingness to engage with services. This article deepens the theoretical and empirical foundations of the field through four contributions: (1) an expanded literature review integrating privacy-by-design, policy and architecture approaches with contemporary acceptance frameworks (TAM/UTAUT) and privacy calculus; (2) a qualitative methodology with transparent sampling, instrumentation, and hybrid thematic analysis; (3) cross-service comparisons across wearables, mobile health apps, telehealth, and medicine delivery that explain when and why users trade privacy for utility; and (4) prescriptive design guidelines for perceptible privacy, granular control, and adaptive authentication. We identify three themes: disadvantaged user position, privacy reassurance, and user-experience barriers, and show that perceived transparency and perceived control are proximal determinants of trust, distinct from generic privacy risk. We propose interface-level interventions including access-log visibility, plain-language policy twins, and risk-tiered MFA that increase willingness to engage without sacrificing safety. These findings call for perceived transparency to be treated as a distinct and measurable construct in health technology adoption models, alongside privacy risk.**

Keywords-smart healthcare; privacy protection measures; user perceptions; perceived transparency; perceived control; privacy calculus; TAM; UTAUT; qualitative methods; NVivo; adaptive authentication.

I. Introduction

Smart cities embody the integration of digital technologies into urban systems, with smart healthcare emerging as a key sector [2]. By leveraging sensors, Internet of Things (IoT) devices, and data analytics, smart healthcare aims to enhance service delivery and quality of life, particularly in response to urbanization challenges such as population growth [1,2,3,4]. Users are central to this ecosystem, both as data contributors and service beneficiaries. However, the diverse and sensitive nature of the data collected, particulary personal health data—necessitates robust privacy protection. Healthcare has become a primary target for data breaches, leading to heightened privacy concerns and user avoidance behaviors that hinder adoption and effectiveness [5,6,7,8,9,10,11]. This study focuses on users' perceptions of privacy protection in smart healthcare, particularly in terms of technical (e.g., encryption) and organizational (e.g., privacy policies) measures. Given their role as data owners, patients' engagement is vital to the success of smart healthcare services [12,13]. However, despite the availability of privacy-preserving technologies like blockchain, homomorphic encryption, and secure multi-party computation, users often lack awareness or confidence in these tools. The effectiveness of such measures is influenced not only by their technical strength but also by users' psychological perceptions of security [10,13,14,15].

User perceptions—shaped by factors such as perceived control, information risk, and expected societal benefits—significantly influence their willingness to disclose data and use smart healthcare services [16,17,18]. Technology acceptance models like UTAUT have been used to explore these dynamics, showing that while users recognize the benefits of digital health services, privacy and trust issues remain critical barriers to adoption [19,20]. These concerns are not just technical but deeply human, highlighting the need to bridge the gap between system design and user expectations. Despite growing attention, many studies still overlook the user's perspective on privacy protection. Limited awareness, passive consent, and a lack of empowerment persist due to the unequal power dynamics between users and service providers [13,22,23]. Effective communication of data protection measures is lacking, preventing users from making informed privacy decisions [24,25]. To advance smart healthcare adoption, future systems must prioritize transparency, user education, and privacy frameworks that align with user preferences and perceptions. This study aims to contribute by deepening the understanding of these user-centered concerns and informing more inclusive privacy strategies. To help users better protect and control their privacy, it is essential to improve the communication of privacy protection measures to users and raise their privacy awareness. The first step in this process is ensuring they perceive the privacy measures in place. Thus, it is crucial to address the research problem underlying this paper, namely, to understand users' perceptions of privacy protection measures in smart healthcare services. The paper is organized as follows. Section II outlines related work, theoretical foundations, and the research methodology. Section III presents the results, Section IV discusses their implications, and Section V concludes with key insights and future directions.

R.Q: How do users perceive privacy protection measures in smart healthcare services? The study aims to explore users' concerns and expectations when perceiving privacy protection measures in smart healthcare services. Understanding users' nuanced feelings is essential for designing privacy safeguards that are user-oriented, ensuring better alignment with users' privacy needs. Relative to the conference predecessor [1], this

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

journal article substantially extends the theoretical framing, details the methodological procedures, adds cross-service comparative analysis, and contributes prescriptive design guidelines — content that was absent from the original.

II. Related Work and Theoretical Framework

To position our study in a fuller landscape, we now detail how organizational and technical privacy measures intersect with user acceptance and privacy calculus, and motivate why "perceived transparency" merits elevation as a distinct construct in health technology adoption.

A. Organizational Privacy Approaches and Regulation

Organizational privacy approaches—privacy-by-policy, privacy-by-architecture, and privacy-by-design—provide complementary routes for embedding privacy into smart healthcare systems. In practice, these approaches materialize through consent management, transparency requirements, auditing, and accountability mechanisms mandated by regulatory regimes such as the GDPR and HIPAA, which shape both provider incentives and users' trust heuristics. However, the findings in this study—building on the conference predecessor [1]—show that users' trust depends less on the existence of these policies and more on the perceivability of their enforcement, such as knowing who accessed their data, how privacy measures were implemented, and whether deletion or minimization actually occurred. This distinction allows us to reconceptualize organizational mechanisms within UTAUT by treating audits and breach-response practices as external facilitating conditions, while policy updates, notifications, and in-product explanations function as internal facilitating conditions, thereby refining the broad and often ambiguous original definition.

B. Technical Measures and Their Perceived Reliability

Beyond encryption, anonymization, access control, and authentication, this study examines differential privacy, homomorphic encryption, SMPC, and federated learning [1]. Security event monitoring and service-level accountability are also relevant dimensions: attack event monitoring in cyber-physical systems [20] and SLA-based monitoring for cloud services [21] demonstrate how continuous oversight mechanisms can reinforce the technical safeguards that underpin smart healthcare infrastructure. However, interviewees' confidence was often moderated by implementation opacity (e.g., key management, re-identification risk). We thus separate technical strength from perceived reliability, the latter rising when providers layer techniques and disclose third-party audits.

C. Acceptance Models with Privacy Calculus—and Why "Perceived Transparency" Matters

TAM (usefulness/ease-of-use) and UTAUT (performance/effort expectancy, social influence, facilitating conditions) explain why strong security alone is insufficient if adoption costs remain high. Privacy calculus explains users' cost–benefit trade-offs; yet this study shows that perceived transparency and perceived control function as proximal determinants of trust and willingness to disclose, distinct from generic privacy risk. We therefore propose a model extension where perceived transparency (who/what/how/with what effect) moderates the link between privacy risk and trust, particularly in high-stakes services (telehealth, prescriptions)

III. Research Baseline

Smart healthcare has emerged as a response to the increasing strain on traditional healthcare systems caused by population growth and rising disease prevalence. Leveraging technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and mobile cloud computing, smart healthcare enhances communication, facilitates remote monitoring, and supports personalized treatment, diagnosis, and prevention efforts [26,27]. These services are broadly categorized into domains like location tracking, telehealth, mobile health, AI-driven diagnostics, and robotic systems [28]. Among these, five key application types are emphasized: tracking tools (e.g., Apple Watch), telehealth platforms (e.g., BetterHelp), AI-powered diagnostic systems (e.g., IBM Watson Health), integrated health information systems (e.g., Epic Systems), and medicine delivery platforms (e.g., Amazon Pharmacy).

This study focuses on four types of smart healthcare services—wearable devices, mobile health management apps, telehealth platforms, and medicine delivery systems—due to their widespread user adoption and diverse data handling. These services directly involve users in managing vital signs, lifestyle data, medical records, and prescriptions. For example, wearable devices like Apple Health gather physiological data, while telehealth platforms support virtual consultations. These applications offer close user interaction, in contrast to more provider-centric systems such as Electronic Health Record (EHRs) or AI-assisted surgery, which are excluded from the study's scope. The selected services provide a relevant and practical basis for examining users' perceptions of privacy protection in smart healthcare.

Privacy Protection Measures: Smart healthcare systems face ongoing challenges in safeguarding user privacy throughout the data lifecycle, despite the many benefits they offer [11] [29]. To address these concerns, researchers have proposed various privacy protection strategies, including both technical and organizational measures. Organizational approaches such as privacy-by-policy, privacy-by-architecture, and privacy-by-design aim to embed privacy into system design, policy compliance, and user interactions from the outset [31,32,33].

These are further supported by regulatory frameworks like the GDPR and HIPAA, which enforce strict legal standards for personal health data handling. Organizational mechanisms such as consent management, transparency, and auditing play a key role in maintaining accountability and building trust [3]. Technically, privacy is protected through cryptographic methods, anonymization, data masking, access control, and advanced techniques like federated learning, homomorphic encryption, and secure multi-party computation [34,35]. Blockchain is also recognized for its privacy-enhancing attributes, including

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

decentralization and transparency [6]. However, these measures often fall short in practice due to limited user control and inconsistent implementations across centralized and decentralized environments [24].

Many existing solutions remain too provider-centric, failing to fully address user needs or empower them in managing their own data [6,11]. As such, there is a pressing need for privacy strategies that incorporate users' perspectives more effectively and bridge the gap between technical safeguards and user-centric privacy experiences. Smart healthcare privacy is co-produced by organizational regimes and technical safeguards. Organizationally, privacy-by-policy, privacy-by-architecture, and privacy-by-design approaches embed protections into policies, system boundaries, and engineering workflows.

Under GDPR and HIPAA, they manifest as consent management, data minimization, storage limitation, purpose specification, auditability, and breach notification. Yet interviews in our study repeatedly showed that users rarely trust a policy merely because it exists; they trust it when it becomes perceptible—for example, when a policy statement that "access is restricted" is matched by a user-visible access log, or when "deletion on request" is accompanied by a deletion receipt and partner-propagation status. These findings complement prior literature emphasizing technical controls, by highlighting the human salience of enforcement visibility. Organizational measures thus serve a dual role: they constrain providers and, when made visible, reassure users. Analogous accountability mechanisms in adjacent domains — such as SLA monitoring for cloud services [21] and security event monitoring in cyber-physical systems [20] — illustrate how continuous oversight translates technical guarantees into verifiable, user-facing assurances.

On the technical side, encryption, anonymization, access control, and authentication remain foundational. Participants widely supported multi-layer protection, not only because it reduces single points of failure, but because each layer can be communicated to users in different ways (e.g., lock icons for encrypted sessions, mask icons for anonymized records). Emerging techniques such as differential privacy, homomorphic encryption, secure multi-party computation, and federated learning extend protections into new computational settings; however, user confidence hinged on implementation clarity—where keys live, how re-identification is prevented, and what third-party validation exists. This suggests a distinction between technical strength and perceived reliability: the latter improves when providers publish audit summaries, use plain-language explanations with examples, and provide outcome proofs (e.g., deletion confirmations). Classic acceptance frameworks help explain these reactions. TAM and UTAUT predict that even strong safeguards will not drive adoption if the system is hard to use; our interviews show how cognitive (policy readability) and operational (authentication friction) barriers map directly to effort expectancy and perceived ease of use. Privacy calculus clarifies why some users accept risk in high-necessity contexts (telehealth) yet refuse to share lifestyle data in non-essential apps. Our data point to perceived transparency—clarity about who accesses data, how protections operate, and what results they produce—as a key mediator between privacy risk and trust. We therefore propose to treat perceived transparency alongside perceived control as distinct and measurable constructs in health technology adoption models, especially for high-stakes contexts.

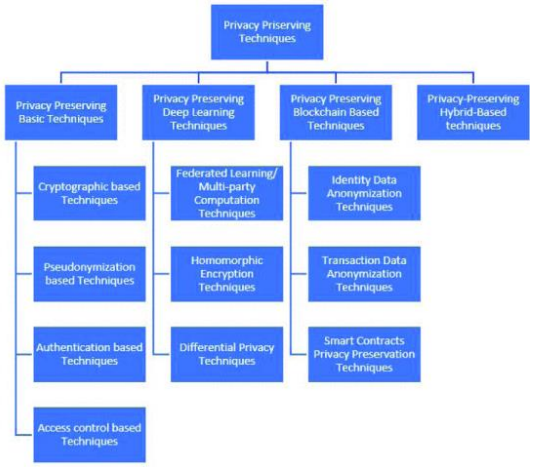


Figure 1. Privacy-preserving Techniques Taxonomy [34]

Theories and Factors Shaping Users' Perception and Adoption: Users' perceptions significantly influence their intention to adopt smart healthcare services, as outlined in established frameworks like the Technology Acceptance Model (TAM) and the Unified Theory of Acceptance and Use of Technology (UTAUT) [36,37]. TAM focuses on perceived usefulness and ease of use, while UTAUT highlights performance expectancy, effort expectancy, social influence, and facilitating conditions. These models are further enriched by the privacy calculus theory, which balances perceived benefits against privacy risks [38,39]. Research shows that risks such as data misuse, legal vulnerabilities, and lack of control affect users' protective behaviors, while perceived benefits like better healthcare and contributions to research often encourage data sharing [9,40]. Emotions, cognitive biases, and contextual factors also shape decision-making.

Although privacy remains a concern, users often prioritize perceived benefits, especially when immediate rewards or limited privacy knowledge come into play [41,42,43]. Older adults, for instance, tend to accept privacy risks over time, leading to a resigned attitude toward potential misuse [44]. For wearables, adoption hinges on a risk-benefit evaluation tied to data sensitivity and regulatory protections [45]. However, challenges like opaque privacy notices and cognitive overload hinder informed decisions [46,47,48]. Emerging factors—such as trust in AI, personalization, and digital literacy—further affect adoption, prompting scholars to recommend tailoring acceptance models to specific healthcare contexts [49,50,51,52]. This study adopts relevant theoretical constructs to capture the nuanced dimensions of user perception in smart healthcare.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A. Method Application

We adopted a qualitative, interview-based survey to elicit nuanced perceptions that structured questionnaires struggle to surface. The semi-structured instrument (40 items; Appendix B) linked each question to at least one construct from TAM, UTAUT, privacy calculus, or a taxonomy of privacy-preserving techniques, thereby enhancing content validity. Interviews were 60–90 minutes (one-to-one via WhatsApp/WeChat), audio-recorded with participant consent, and transcribed using Voice Memos (English) and Tongyi.ai (Chinese), then manually checked against the audio and validated by participants.

Sampling employed purposive logic to recruit adults aged 18–65 in urban settings across Europe, North America, and Asia who reported prior experience with smart healthcare and baseline privacy awareness. This choice privileged information-rich cases over randomness to meet the study aim. We anticipated saturation near 10–12 interviews and observed redundancy by the tenth interview, consistent with qualitative sampling guidance; participant demographics appear in Appendix A.

Analysis followed Braun and Clarke's six-step thematic procedure in NVivo. We first open-coded transcripts (21 initial codes), then consolidated semantically overlapping codes (e.g., "Unknown Technical Principles" → "Unknown Implementation Process"), and organized them into 12 refined codes mapped to three themes and seven subthemes. To temper researcher subjectivity, deductive lenses from TAM/UTAUT/privacy calculus were applied secondarily when reviewing and naming themes, while honoring inductively emergent content. We tracked all merges/splits for auditability and present final theme/subtheme/code mappings in Table I.

Ethics emphasized data minimization and participant rights. We collected no specific health data; demographics were coarse-grained (age bands, not exact ages; industry, not job title). Audio files were stored on encrypted devices and could be deleted at participant request after transcript verification; transcripts were anonymized and access-restricted, aligning with GDPR principles and erasure rights.

This study employs a qualitative, interview-based survey strategy to explore users' perceptions of privacy protection in smart healthcare services. Qualitative surveys effectively reveal nuanced behaviors, perceptions, and attitudes, making them particularly suitable for understanding complex interactions between privacy risks, benefits, and other factors [53,54,56]. To ensure rigor and reliability, this study integrates standardized theoretical frameworks, including the taxonomy of privacy-preserving techniques [34], Technology Acceptance Model (TAM), Unified Theory of Acceptance and Use of Technology (UTAUT), and privacy calculus theory, to guide interview design and analysis [55]. These frameworks help structure the interview questions effectively by addressing key constructs such as perceived ease of use, social influence, performance expectancy, perceived usefulness, perceived privacy risks, and perceived benefits. Interviews are conducted individually through audio calls on WhatsApp and WeChat, recorded and transcribed using the iOS Voice Memos application for English and Tongyi.ai for Chinese interviews. Recordings are solely for transcription accuracy and verification purposes, ensuring data reliability. **Sampling:**This study employs non-probability purposive sampling, deliberately selecting participants aged 18–65 from urban areas in Europe, North America, and Asia who possess prior experience with smart healthcare services and basic privacy protection knowledge, ensuring their relevance to the research topic [57,58]. Probability sampling was not chosen due to the need for informed participants rather than random selection. Ethical considerations and resource constraints excluded minors, individuals over 65, and rural populations, as these groups pose consent challenges or may lack familiarity with smart healthcare [13]. The target sample size is approximately 10 participants, consistent with qualitative research standards indicating saturation typically occurs between 10 to 12 interviews [60,61]. Interviews lasting 60–90 minutes ensure comprehensive coverage of key points, enhancing study validity despite the limited sample size. Convenience sampling was dismissed due to its potential limitations in participant diversity and relevance [57,59]. **Data Analysis:** The study analyzes interview data using Thematic Analysis (TA), a widely adopted qualitative method suitable for identifying patterns and themes within interview transcripts [62,63]. Following the six-step process outlined by Braun and Clarke [64,66], the analysis begins by coding relevant keywords, refining codes to eliminate redundancy, and categorizing them into themes. This study adopts a hybrid thematic approach, primarily utilizing inductive thematic analysis to allow patterns to emerge naturally, complemented by deductive analysis based on constructs from the Technology Acceptance Model (TAM), Unified Theory of Acceptance and Use of Technology (UTAUT), and privacy calculus theory to enhance objectivity and validity [65,66,67,68]. NVivo software is employed to improve analytical efficiency, data management, and accuracy [69,70]. Content analysis was considered but not selected due to its limited ability to capture contextual and latent meanings critical to the research [57,71,72].

IV. Result

This study applied thematic analysis supported by NVivo software, following the structured six-step process outlined by [64,66]. Initial coding involved careful review of interview transcripts, capturing significant insights and relating these to theoretical frameworks. Notably, participant uncertainties, such as limited knowledge about privacy protection, were documented. The code "Cumbersome Authentication Process" drew upon "perceived ease of use" from the TAM and "effort expectancy" from the Unified Theory of Acceptance and Use of Technology (UTAUT). In total, 21 initial codes emerged. From these codes, themes and subthemes were developed, categorizing user perceptions into positive and negative dimensions. A prominent theme identified was the "Disadvantaged Position of Users in Protecting Their Privacy," with the subtheme "Insufficient Right to be Informed," consistently emphasized by participants. Conversely, positive perceptions were encapsulated under the theme related to "facilitating conditions" from

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

UTAUT, distinguishing between external support (Oversight and Constraints) and internal support (Internal Handling). Redundant codes were subsequently refined, reducing them to 12 cohesive codes. For instance, "Unknown Technical Principles" merged into "Unknown Implementation Process," reflecting users' practical concerns rather than theoretical knowledge. Similarly, "Insufficient Engagement" was incorporated into "Lack of User Data Management," categorized under the subtheme "Insufficient Control." Ultimately, the refined analysis identified three main themes and seven subthemes from the dataset.

A. The Disadvantaged Position of Users in Protecting Their Privacy

Users' Perceived Vulnerability in Privacy Protection Thematic analysis revealed users feel disadvantaged in safeguarding their privacy within smart healthcare services, reflecting concerns from privacy calculus theory. Despite existing measures, users often lack perceived control and awareness, particularly in three key areas: unknown personnel, unclear implementation processes, and uncertain effectiveness of results. Users expressed concerns about not knowing who accesses their data, especially with vague or opaque privacy policies and multi-party access scenarios. Many feared unauthorized third-party data sharing, particularly for commercial purposes, while showing more openness toward research uses, provided transparency and consent are maintained.

Lack of Transparency Undermines Trust: Participants emphasized that unfamiliar technical implementations (e.g., encryption, anonymization) raise doubts, especially when not clearly explained. Even tech-savvy users sought clarity on compatibility and deployment, while others feared such terms masked hidden costs or misuse. Vague legal language in privacy policies also contributed to uncertainty about data handling. Users wanted clear, example-driven explanations of practices and desired features like access logs, deletion confirmation, and visual cues for encryption [74]. Ultimately, the study highlights how insufficient transparency undermines users' sense of security, making trust in privacy protections contingent on clarity, informed consent, and demonstrable effectiveness.

Users' Limited Control and Data Deletion Challenges: Participants expressed a strong sense of limited control over their data within smart healthcare services, often feeling reliant on providers who possess technical knowledge and control system configurations. This asymmetry reinforces user vulnerability, as providers determine how and when data is used [46]. While participants desired more autonomy—such as opt-in/out capabilities for data use, the ability to delete records post-service, and clearer management of access rights—these features remain inadequately supported. Even in GDPR-covered regions, deletion processes are often slow, indirect, or poorly designed, further disempowering users. Participants from outside the GDPR context reported even fewer deletion options, highlighting global inconsistencies in privacy control.

Inadequate Consent and Forced Privacy Agreements: Granular consent management was viewed as essential, with users preferring settings that allow them to specify the purpose, scope, and recipients of shared data. However, participants described being forced into "take it or leave it" agreements during account registration, where refusing a privacy policy meant losing access to the service entirely [47]. This coercive design fosters mistrust in both the policies and the providers themselves, though it doesn't always deter usage, especially when the service is deemed necessary. Participants emphasized that privacy policies often serve as compliance tools rather than genuine efforts to respect user preferences [5,47]. To restore meaningful control, users should be empowered to use core services even if they partially or fully reject privacy terms.

B. Privacy Reassurance

Legal Compliance as a Source of Reassurance: Participants expressed generally positive views toward privacy regulations like the GDPR and HIPAA, associating compliance with increased confidence in smart healthcare services. GDPR compliance, in particular, was seen as a strong indicator of trustworthy data practices due to its well-defined principles, independent oversight, and strict penalties for violations . Some participants emphasized that GDPR offers not only legal assurance but also actionable tools for users to verify compliance and seek redress. In contrast, while HIPAA was acknowledged for setting essential standards, American participants showed relatively lower confidence in its enforcement and practical application. This suggests that users value legal frameworks more when they are backed by demonstrable enforcement mechanisms and transparent rights protections [73].

The Role of Audits in Reinforcing Trust: External audits were highlighted as a crucial organizational safeguard complementing legal compliance. Participants emphasized that while privacy-enhancing technologies like encryption and anonymization are important, their trust increases when these measures are validated through transparent third-party audits. Audits conducted by reputable firms enhance users' perception of provider accountability, particularly because users often lack the expertise to evaluate technical safeguards themselves. Together, legal constraints and professional audits offer layered protection that aligns with the UTAUT framework's notions of performance expectancy and facilitating conditions, reinforcing users' belief that their privacy is both respected and technically safeguarded.

Multi-layer Protection and Technical Confidence Participants expressed strong support for multi-layer privacy protection, noting that the combination of various techniques—such as encryption, anonymization, and multi-factor authentication—enhanced their trust in smart healthcare systems. While many users lacked technical expertise, they believed that layering different methods could reduce single points of failure and increase reliability. Features like two-factor authentication were especially appreciated, as they offered visible, user-facing indicators of security. This sense of reassurance directly influenced users' willingness to engage with smart healthcare services, aligning with the UTAUT construct of performance expectancy.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. THEMES, SUBTHEMES, AND CORRESPONDING CODES

Theme	Subtheme	Codes
The Disadvantaged Position of Users in Protecting Their Privacy	Insufficient Right to be Informed	Unknown Personnel Involved Unknown Implementation Process Unknown Effectiveness of Results
	Insufficient Control	Lack of User Data Management Passive Choice in Privacy Policy
Privacy Reassurance	Oversight and Constraints	Legal Regulations and Audits
	Technical Reliability	Multi-Layer Protection
	Internal Handling	Updates of Privacy Policy Thoughtful Data Breach Response
User Experience Barriers	Operational Barriers	Cumbersome Authentication Process Unclear Position of the Privacy Policy
	Cognitive Barriers	Long and Obscure Privacy Policy

Proactive Provider Measures and Breach Response Users also valued internal organizational practices, such as timely updates to privacy policies and responsive actions following data breaches, as signs of a provider's commitment to privacy protection. Regular policy updates, when clearly communicated, reassured participants that providers were keeping pace with technological and legal changes. After a breach, participants expected prompt notifications, transparency about affected data, and evidence of corrective action—such as improved security systems or audits. These proactive and reflective efforts serve as key facilitating conditions that influence continued user trust, even after a privacy incident. Providers who effectively communicate updates and breach responses are more likely to retain user confidence in the long term.

Cognitive Barriers to Understanding Privacy Policies: Participants widely reported cognitive challenges when engaging with privacy policies, citing long, text-heavy documents and complex legal jargon as key deterrents to reading or understanding them. These barriers were especially burdensome for older users, who also faced physical and digital literacy limitations. The confusing presentation and obscure terminology led to user frustration and mistrust, with some perceiving the complexity as an intentional obfuscation by providers. Participants suggested clearer formats like visual checklists, interactive summaries, and plain language versions to improve comprehension and enhance trust. Offering two parallel policy versions—one simplified and one legally detailed—was proposed to balance accessibility with compliance requirements.

Operational Barriers and Their Contextual Impact: Operational hurdles, particularly around authentication processes, were another major concern. Users found complex password requirements and recovery procedures burdensome, especially when compounded by poor connectivity or urgent health needs. While users accepted stricter authentication for high-risk services like mental health or prescriptions, they preferred minimal friction for lower-risk tasks like symptom checking or step tracking. Participants also noted difficulty locating privacy policies within app interfaces, which undermined their perceived importance. Although this didn't always affect service use directly, it shaped negative impressions of provider transparency. Users emphasized the need for adaptive privacy measures and intuitive design that aligns security requirements with task sensitivity and user context.

V. DESIGN GUIDELINES AND PRACTITIONER PLAYBOOK

Beyond explaining perceptions, we translate themes into implementable patterns providers can adopt.

A. Make Protections Perceptible

- **Access-log visibility:** Provide user-readable logs showing who accessed their data, when the access occurred, and what information was viewed. Include options for exporting logs and issuing anomaly alerts.
- **Inline encryption cues:** Surface end-to-end encryption at the point of use, such as displaying a message like "Only you and your clinician can view this chat".
- **Deletion receipts:** Offer immediate confirmation of data deletion through on-screen messages and follow-up email or SMS notifications. Display whether deletions have propagated to partner systems or backups.

B. Restore User Agency

- **Plain-language policy twin:** Provide a user-friendly privacy policy written in clear language with examples, fully searchable and cross-linked to the corresponding legal version.
- **Granular consent dashboard:** Allow users to control purpose, scope, and retention of data processing through toggle-based settings, with defaults adjusted according to service risk.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

- **Partial-service access:** Enable continued access to essential functions even when certain policy terms are partially refused, supporting user autonomy in sensitive healthcare contexts.

C. Optimize Friction with Adaptive Authentication

- **Risk-tiered MFA:** Trigger multi-factor authentication for high-risk actions such as sharing medical records, modifying prescriptions, or completing payments, while keeping access to low-risk information quick and unobstructed.
- **Context signals:** Adapt authentication to situational factors—for example, offering offline one-time codes or deferring heavy verification steps when network connectivity is poor, while maintaining an audit trail.
- **Biometrics with fallbacks:** Support user-consented biometric authentication (e.g., facial or fingerprint recognition), along with password manager guidance or fallback options when biometrics are unavailable.

D. Educate at the Moment of Risk

- **Micro-nudges:** Present short animations, prompts, or tooltips when users engage in potentially risky actions, such as attempting to share data with an unverified party.
- **Color-coded risk labels:** Use intuitive red/amber/green indicators accompanied by plain-language explanations to help users understand different risk levels without relying on technical jargon.

Participants calibrated privacy expectations to the perceived risk of each service: telehealth was deemed most sensitive due to detailed medical history, wearables were moderated by brand reputation and social endorsement, mobile health apps were tolerated when the benefit was clear but often rejected for intrusive lifestyle tracking, and medicine delivery raised unique physical-world safety concerns (home address visibility, in-home interactions). These perceptions help explain why a one-size authentication policy fails: users readily accept MFA and extra checks when sharing a medical report, but resent the same friction when glancing at steps or weight. Our interviews thus argue for risk-tiered defaults: stringent consent and audit trails for telehealth and delivery, and streamlined but still safe interactions for low-risk app features.

VI. Discussion

A. User-Centered Design Patterns

We propose four user-centered design patterns that support more transparent, controllable, and trustworthy smart healthcare interactions.

1) **Perceptible privacy.** Surface privacy protections at the moment of use, such as end-to-end encryption badges in chat interfaces or mask icons on anonymized data fields. Provide users with readable access logs and verifiable deletion receipts, and publish short audit summaries that communicate the effectiveness of implemented safeguards.
2) **Real control.** Offer a plain-language "policy twin" that includes examples and is cross-linked to the full legal version of the privacy policy. Provide a granular consent dashboard that allows users to adjust purpose, scope, and retention options. Ensure that essential service features remain accessible even when certain secondary data uses are refused.
3) **Adaptive authentication.** Align authentication strength with the level of risk. Trigger multi-factor authentication for high-stakes actions such as sharing medical records, modifying prescriptions, or conducting payments. Enable biometric authentication where appropriate and provide robust offline fallback mechanisms.
4) **Just-in-time education.** Present timely micro-nudges and intuitive color-coded risk indicators at the moment users attempt potentially harmful actions. These cues should be paired with short explanations and quick-access links to relevant privacy or security settings.

Together, these patterns enhance *perceived transparency* and *perceived control*, the two user-experience factors most strongly associated with trust and willingness to disclose sensitive information.

This study reveals that users generally view privacy protection measures in smart healthcare positively, particularly when multi-layer safeguards—such as encryption, anonymization, and multi-factor authentication—are employed, reinforcing their sense of security and aligning with UTAUT's performance expectancy construct. However, users also expressed concerns about limited transparency and control, especially when faced with complex or opaque privacy policies. Legal frameworks like the GDPR and HIPAA, along with third-party audits, were seen as crucial external supports that help balance the power disparity between users and providers [30,39]. Despite recognizing these safeguards, users often felt disempowered due to their lack of technical or legal literacy, particularly in urgent health situations where privacy is traded for immediate care needs [9,22,43].

These tensions echo the privacy calculus theory, where perceived privacy risks reduce trust and adoption willingness [45], though this is sometimes overridden by brand trust or social influence [19]. Users' perceptions of privacy risk vary based on the type of smart healthcare service and the sensitivity of data involved. Telehealth platforms were seen as higher risk due to their handling of sensitive medical histories, while wearable devices and mobile health apps were perceived as lower risk depending on context [9,41,42]. Unique concerns were also raised about medicine delivery services, particularly involving the disclosure of home addresses in offline interactions.

These findings emphasize that privacy protections must be contextually adaptive. The study contributes to the literature by highlighting the often-overlooked user perspective, suggesting refinements to existing models like UTAUT and the privacy calculus theory, and offering actionable recommendations to enhance transparency, control, and user empowerment in smart healthcare design.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

VII. CONCLUSIONS AND FUTURE WORK

This article moves the discussion from which measures exist to which measures users can see, understand, and control. We show that trust in smart healthcare is constructed through the visibility of process and proof of outcomes, not only the underlying technical strength. A refined model that elevates perceived transparency and nuances facilitating conditions, distinguishing between external and internal conditions, accounts for why the same technical control can reassure in one context and raise suspicion in another. Practically, we outline four interface-level patterns: perceptible privacy, real control, adaptive authentication, and just-in-time education, all of which align security with human cognition and task needs. Future work should quantitatively validate perceived transparency as a mediator in adoption models and test the proposed design patterns through controlled experiments and field deployments across telehealth, wearables, mobile health apps, and medicine delivery.

VIII. AI USE DECLARATION

The authors conceived, designed, and conducted all aspects of this research, including the study framework, interview design, data collection, thematic analysis, and design guidelines. ChatGPT (OpenAI GPT-4) was used solely as a writing aid for text rephrasing, literature search support, and LaTeX formatting. All outputs were reviewed and validated by the authors. No AI tool contributed to the scientific content or conclusions.

REFERENCES

[1] H. Guo, E. Seid, Y. Li, and F. Blix, "Evaluating User Perceptions of Privacy Protection in Smart Healthcare Services,´´ in *SECURWARE 2025: The Nineteenth International Conference on Emerging Security Information, Systems and Technologies*, 2025.

[2] V. Garcia-Font, "SocialBlock: An architecture for decentralized user-centric data management applications for communications in smart cities," *Journal of Parallel and Distributed Computing*, vol. 145, pp. 13–23, Nov. 2020. doi: 10.1016/j.jpdc.2020.06.004

[3] V. Zimmermann, "Smart cities as a testbed for experimenting with humans? - Applying psychological ethical guidelines to smart city interventions," *Ethics and Information Technology*, vol. 25, no. 4, p. 54, Oct. 2023. doi: 10.1007/s10676-023-09729-3

[4] D. Eckhoff and I. Wagner, "Privacy in the Smart City—Applications, Technologies, Challenges, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 489–516, 2018. doi: 10.1109/COMST.2017.2748998

[5] J. Sanghavi, "Review of Smart Healthcare Systems and Applications for Smart Cities," in *ICCCE 2019*, A. Kumar and S. Mozar, Eds. Singapore: Springer Singapore, 2020, pp. 325–331.

[6] European Union, "General Data Protection Regulation (GDPR)," 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679

[7] D. El Majdoubi, H. El Bakkali, S. Sadki, Z. Maqour, and A. Leghmid, "The Systematic Literature Review of Privacy-Preserving Solutions in Smart Healthcare Environment," *Security and Communication Networks*, vol. 2022, no. 1, p. 5642026, 2022. doi: 10.1155/2022/5642026

[8] Y. Li, "Empirical Studies on Online Information Privacy Concerns: Literature Review and an Integrative Framework," *Communications of the Association for Information Systems*, vol. 28, pp. 453–496, Jan. 2011. doi: 10.17705/1CAIS.02828

[9] S. R. Simon, J. S. Evans, A. Benjamin, D. Delano, and D. W. Bates, "Patients' Attitudes Toward Electronic Health Information Exchange: Qualitative Study," *Journal of Medical Internet Research*, vol. 11, no. 3, p. e30, 2009. doi: 10.2196/jmir.1164

[10] X. Deng, D. Wang, and L. Yang, "The Impact of Perceived Risk on Online Medical Users' Privacy Protection Behavior," in *Proc. 27th Int. Conf. on Computer Supported Cooperative Work in Design (CSCWD)*, May 2024, pp. 1238–1243. doi: 10.1109/CSCWD61410.2024.10580280

[11] A. Odeh, A. Eman, and S. Walid, "Privacy-Preserving Data Sharing in Telehealth Services," *Applied Sciences*, 2024. doi: 10.3390/app142310808

[12] M. A. Sahi *et al.*, "Privacy Preservation in e-Healthcare Environments: State of the Art and Future Directions," *IEEE Access*, vol. 6, pp. 464–478, 2018. doi: 10.1109/ACCESS.2017.2767561

[13] D. E. Majdoubi, H. E. Bakkali, S. Sadki, A. Leghmid, and Z. Maqour, "HOPPy: Holistic Ontology for Privacy-Preserving in Smart Healthcare environment," in *Proc. 2021 Fifth World Conf. on Smart Trends in Systems Security and Sustainability (WorldS4)*, 29–30 July 2021, pp. 248–253. doi: 10.1109/WorldS451998.2021.9514051

[14] F. Tazi, A. Nandakumar, J. Dykstra, P. Rajivan, and S. Das, "SoK: Analyzing Privacy and Security of Healthcare Data from the User Perspective," *ACM Trans. Comput. Healthcare*, vol. 5, no. 2, p. Article 11, 2024. doi: 10.1145/3650116

[15] C.-L. Hsu and M.-R. Lee, "User acceptance of a community-based healthcare information system preserving user privacy," in *Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life: Proc. 7th Int. Conf. UAHCI 2013, Part III*, Las Vegas, USA, July 21–26, 2013. Springer, pp. 453–462.

[16] S. M. E. Sepasgozar, S. Hawken, S. Sargolzaei, and M. Foroozanfa, "Implementing citizen centric technology in developing smart cities: A model for predicting the acceptance of urban technologies," *Technological Forecasting and Social Change*, vol. 142, pp. 105–116, 2019. doi: 10.1016/j.techfore.2018.09.012

[17] E. M. Schomakers, C. Lidynia, and M. Ziefle, "Listen to My Heart? How Privacy Concerns Shape Users' Acceptance of e-Health Technologies," in *2019 Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 21–23 Oct. 2019, pp. 306–311. doi: 10.1109/WiMOB.2019.8923448

[18] E. Seid, O. Popov, and F. Blix, "Security Attack Behavioural Pattern Analysis for Critical Service Providers,´´ *Journal of Cybersecurity and Privacy*, vol. 4, no. 1, pp. 55–75, 2024.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[19] A. Kharlamov, R. Hohmann, and G. Parry, "Data sharing decisions: Perceptions and intentions in healthcare," *Strategic Change*, vol. 32, no. 6, pp. 223–237, 2023.

[20] E. Seid, O. Popov, and F. Blix, "Towards Security Attack Event Monitoring for Cyber Physical-Systems,´´ in *Proc. 9th Int. Conf. on Information Systems Security and Privacy (ICIISSP 2023)*, Lisbon, Portugal, Feb. 2023, pp. 723–733.

[21] E. Seid, M. Nazifa, S. Gupta, O. Popov, and F. Blix, "Security and SLA Monitoring for Cloud Services,´´ Department of Computer and Systems Sciences, Stockholm University, Sweden.

[22] E. Princi and N. C. Krämer, "Out of control – privacy calculus and the effect of perceived control and moral considerations on the usage of IoT healthcare devices," *Frontiers in Psychology*, vol. 11, p. 582054, 2020.

[23] Y.-J. Moon and Y.-H. Hwang, "A Study of Effects of UTAUT-Based Factors on Acceptance of Smart Health Care Services," in *Advanced Multimedia and Ubiquitous Engineering*, J. J. Park, H.-C. Chao, H. Arabnia, and N. Y. Yen, Eds. Berlin, Heidelberg: Springer, 2016, pp. 317–324.

[24] E. Pouyan, "The Impacts of the Perceived Transparency of Privacy Policies and Trust in Providers for Building Trust in Health Information Exchange: Empirical Study," *JMIR Medical Informatics*, vol. 7, 2019. doi: 10.2196/preprints.14050

[25] K. Halvorsen *et al.*, "Empowerment in healthcare: A thematic synthesis and critical discussion of concept analyses of empowerment," *Patient Education and Counseling*, vol. 103, no. 7, pp. 1263–1271, Jul. 2020. doi: 10.1016/j.pec.2020.02.017

[26] M. Duckert and L. Barkhuus, "Protecting Personal Health Data through Privacy Awareness: A study of perceived data privacy among people with chronic or long-term illness," *Proc. ACM Hum.-Comput. Interact.*, vol. 6, no. GROUP, p. Article 11, 2022. doi: 10.1145/3492830

[27] M. N. Alraja, H. Barhamgi, A. Rattrout, and M. Barhamgi, "An integrated framework for privacy protection in IoT — Applied to smart healthcare," *Computers & Electrical Engineering*, vol. 91, p. 107060, May 2021. doi: 10.1016/j.compeleceng.2021.107060

[28] S. M. Williamson and V. Prybutok, "Balancing Privacy and Progress: A Review of Privacy Challenges, Systemic Oversight, and Patient Perceptions in AI-Driven Healthcare," *Applied Sciences*, vol. 14, no. 2, p. 675, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/2/675

[29] S. S. Raoof and M. A. S. Durai, "A Comprehensive Review on Smart Health Care: Applications, Paradigms, and Challenges with Case Studies," *Contrast Media & Molecular Imaging*, vol. 2022, no. 1, p. 4822235, 2022. doi: 10.1155/2022/4822235

[30] M. A. Jabbar, K. M. V. V. Prasad, and R. Aluvalu, "Reimagining the Indian Healthcare Ecosystem with AI for a Healthy Smart City," in *Emerging Technologies in Data Mining and Information Security*, A. E. Hassanien, S. Bhattacharyya, S. Chakrabati, A. Bhattacharya, and S. Dutta, Eds. Singapore: Springer, 2021, pp. 543–551.

[31] H. Kwon *et al.*, "Review of smart hospital services in real healthcare environments," *Healthcare Informatics Research*, vol. 28, no. 1, pp. 3–15, 2022.

[32] A. Algarni, "A Survey and Classification of Security and Privacy Research in Smart Healthcare Systems," *IEEE Access*, vol. 7, pp. 101879–101894, 2019. doi: 10.1109/ACCESS.2019.2930962

[33] H. Liu, X. Yao, T. Yang, and H. Ning, "Cooperative Privacy Preservation for Wearable Devices in Hybrid Computing-Based Smart Health," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1352–1362, 2019. doi: 10.1109/JIOT.2018.2843561

[34] S. Spiekermann and L. F. Cranor, "Engineering Privacy," *IEEE Transactions on Software Engineering*, vol. 35, no. 1, pp. 67–82, 2009. doi: 10.1109/tse.2008.88

[35] A. A. Alghanim, S. M. M. Rahman, and M. A. Hossain, "Privacy Analysis of Smart City Healthcare Services," in *Proc. 2017 IEEE Int. Symp. on Multimedia (ISM)*, 11–13 Dec. 2017, pp. 394–398. doi: 10.1109/ISM.2017.79

[36] C. Montes *et al.*, "A flexible, privacy enhanced and secured ICT architecture for a smart grid project with active consumers in the city of Zwolle—NL," in *22nd Int. Conf. and Exhibition on Electricity Distribution (CIRED 2013)*, IET, 2013, pp. 1–4.

[37] B. R. Louassef and N. Chikouche, "Privacy preservation in healthcare systems," in *Proc. 2021 Int. Conf. on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP)*, 20–21 Nov. 2021, pp. 1–6. doi: 10.1109/AI-CSP52968.2021.9671083

[38] A. Dangi and R. Mogili, "Privacy Preservation Measure using t-closeness with combined l-diversity and k-anonymity," *Int. J. of Advanced Research in Computer Science and Electronics Engineering*, vol. 1, pp. 28–33, 2012.

[39] A. A. AlQudah, M. Al-Emran, and K. Shaalan, "Technology Acceptance in Healthcare: A Systematic Review," *Applied Sciences*, vol. 11, no. 22, p. 10537, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/22/10537

[40] S. Attuquayefio and H. Addo, "Review of studies with UTAUT as conceptual framework," *European Scientific Journal*, vol. 10, no. 8, 2014.

[41] E.-M. Schomakers, C. Lidynia, and M. Ziefle, "The Role of Privacy in the Acceptance of Smart Technologies: Applying the Privacy Calculus to Technology Acceptance," *Int. J. of Human–Computer Interaction*, vol. 38, no. 13, pp. 1276–1289, 2022. doi: 10.1080/10447318.2021.1994211

[42] S. Li *et al.*, "Research on user's highly sensitive privacy disclosure intention in home intelligent health service system: A perspective from trust enhancement mechanism," *DIGITAL HEALTH*, vol. 9, p. 20552076231219444, 2023. doi: 10.1177/20552076231219444

[43] D. Grande, N. Mitra, A. Shah, F. Wan, and D. A. Asch, "Public preferences about secondary uses of electronic health information," *JAMA Intern Med*, vol. 173, no. 19, pp. 1798–1806, Oct. 2013. doi: 10.1001/jamainternmed.2013.9166

[44] M. S. Rahman, "Does Privacy Matter When We are Sick? An Extended Privacy Calculus Model for Healthcare Technology Adoption Behavior," in *Proc. 2019 10th Int. Conf. on Information and Communication Systems (ICICS)*, 11–13 June 2019, pp. 41–46. doi: 10.1109/IACS.2019.8809175

[45] D. Kim, K. Park, Y. Park, and J.-H. Ahn, "Willingness to provide personal information: Perspective of privacy calculus in IoT services," *Computers in Human Behavior*, vol. 92, pp. 273–281, Mar. 2019. doi: https://doi.org/10.1016/j.chb.2018.11.022

[46] G. Fox, ""To protect my health or to protect my health privacy?" A mixed-methods investigation of the privacy paradox," *Journal of the Association for Information Science and Technology*, vol. 71, no. 9, pp. 1015–1029, 2020.

[47] T. Schroeder, M. Haug, and H. Gewald, "Data Privacy Concerns Using mHealth Apps and Smart Speakers: Comparative Interview Study Among Mature Adults," *JMIR Formative Research*, vol. 6, no. 6, 2022. doi: 10.2196/28025

[48] H. Li, J. Wu, Y. Gao, and Y. Shi, "Examining individuals' adoption of healthcare wearable devices: An empirical study from privacy calculus perspective," *Int. J. of Medical Informatics*, vol. 88, pp. 8–17, Apr. 2016. doi: https://doi.org/10.1016/j.ijmedinf.2015.12.010

[49] A. Acquisti *et al.*, "Nudges for Privacy and Security: Understanding and Assisting Users' Choices Online," *ACM Comput. Surv.*, vol. 50, no. 3, p. Article 44, 2017. doi: 10.1145/3054926

[50] F. Schaub, R. Balebako, and L. F. Cranor, "Designing Effective Privacy Notices and Controls," *IEEE Internet Computing*, vol. 21, no. 3, pp. 70–77, 2017. doi: 10.1109/MIC.2017.75

[51] M. W. Vail, J. B. Earp, and A. I. Antón, "An Empirical Study of Consumer Perceptions and Comprehension of Web Site Privacy Policies," *IEEE Trans. on Engineering Management*, vol. 55, pp. 442–454, 2008.

[52] K. Liu and D. Tao, "The roles of trust, personalization, loss of privacy, and anthropomorphism in public acceptance of smart healthcare services," *Computers in Human Behavior*, vol. 127, p. 107026, Feb. 2022. doi: https://doi.org/10.1016/j.chb.2021.107026

[53] B. Watjatrakul, "Intention to use a free voluntary service: The effects of social influence, knowledge and perceptions," *Journal of Systems and Information Technology*, vol. 15, 2013. doi: 10.1108/13287261311328903

[54] R. Holden and B.-T. Karsh, "The Technology Acceptance Model: Its Past and Its Future in Health Care," *Journal of Biomedical Informatics*, vol. 43, pp. 159–172, Aug. 2009. doi: 10.1016/j.jbi.2009.07.002

[55] V. Braun, V. Clarke, E. Boulton, L. Davey, and C. McEvoy, "The online survey as a qualitative research tool," *International Journal of Social Research Methodology*, vol. 24, pp. 641–654, 2020.

[56] Z. N. Ghafar, "Evaluation Research: A Comparative Analysis of Qualitative and Quantitative Research Methods," *Middle East Research Journal of Linguistics and Literature*, 2023.

[57] J. Melegati, K. Conboy, and D. Graziotin, "Qualitative Surveys in Software Engineering Research: Definition, Critical Review, and Guidelines," *IEEE Transactions on Software Engineering*, vol. 50, pp. 3172–3187, 2024.

[58] K. Semyonov-Tal, "Keeping medical information safe and confidential: a qualitative study on perceptions of Israeli physicians," *Israel Journal of Health Policy Research*, vol. 13, no. 1, p. 54, Sep. 2024. doi: 10.1186/s13584-024-00641-9

[59] M. Denscombe, *The Good Research Guide for Small-Scale Social Research Projects*, 7th ed. Maidenhead, England: Open University Press, 2021.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[60] A. E. Mueller and D. L. Segal, "Structured versus semistructured versus unstructured interviews," *The Encyclopedia of Clinical Psychology*, vol. 1, no. 7, 2014.
[61] United Nations Statistical Office, "Provisional guidelines on standard international age classifications," in *Statistical Papers*, New York: United Nations, 1982.
[62] Y. Lu, M. Jian, N. Muhamad, and M. Hizam-Hanafiah, "Data saturation in qualitative research: A literature review in entrepreneurship study from 2004–2024," *Journal of Infrastructure, Policy and Development*, vol. 8, no. 12, p. 9753, 2024.
[63] D. M. Turner-Bowker *et al.*, "Informing a priori sample size estimation in qualitative concept elicitation interview studies for clinical outcome assessment instrument development," *Value in Health*, vol. 21, no. 7, pp. 839–842, 2018.
[64] C. Herzog, C. Handke, and E. Hitters, "Analyzing Talk and Text II: Thematic Analysis," in *The Palgrave Handbook of Methods for Media Policy Research*, H. Van den Bulck, M. Puppis, K. Donders, and L. Van Audenhove, Eds. Cham: Springer International Publishing, 2019, pp. 385–401.
[65] C. Herzog, C. Handke, and E. Hitters, "Analyzing Talk and Text II: Thematic Analysis," 2019. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3068081
[66] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, pp. 77–101, 2006. doi: 10.1191/1478088706qp063oa
[67] S. Elo and H. Kyngäs, "The qualitative content analysis process," *Journal of Advanced Nursing*, vol. 62, no. 1, pp. 107–115, 2008. doi: 10.1111/j.1365-2648.2007.04569.x
[68] J. Fereday and E. Muir-Cochrane, "Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development," *International Journal of Qualitative Methods*, vol. 5, no. 1, pp. 80–92, 2006. doi: 10.1177/160940690600500107
[69] C. H. Saunders *et al.*, "Practical thematic analysis: a guide for multidisciplinary health services research teams engaging in qualitative analysis," *BMJ*, vol. 381, 2023.
[70] K. A. Campbell *et al.*, "Reflexive thematic analysis for applied qualitative health research," *The Qualitative Report*, vol. 26, no. 6, pp. 2011–2028, 2021.
[71] L. Wong, "Data analysis in qualitative research: a brief guide to using NVivo," *Malaysian Family Physician*, vol. 3, no. 1, pp. 14–20, 2008.
[72] M. I. Azeem and N. A. Salfi, "Usage of NVivo software for qualitative data analysis," 2012.
[73] R. Dubinsky, "PNS169 Personal Medical Health Records Regulation in the United States, European Union and Israel," *Value in Health*, vol. 22, pp. S789–S790, 2019.
[74] C. J. Hoofnagle and J. King, "What Californians Understand About Privacy Online," *Available at SSRN 1133075*, 2008. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1133075

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Automated AWS Security Scanning and DORA Article Mapping: Evidence-Based Compliance for Financial Institutions

Elias Seid, Naol Mengistu, Ali Ferzali, Fredrik Blix
Department of Computer and Systems Sciences
Stockholm University, Sweden
E-mail: (elias.seid) @dsv.su.se

***Abstract*—Financial institutions increasingly rely on Amazon Web Services (AWS) for critical operations, yet cloud misconfigurations remain a leading cause of security breaches and regulatory non-compliance. This paper presents a reproducible, compliance-aware scanning model that programmatically inspects AWS services—S3, EC2, IAM, and VPC—using a Python/Boto3 toolchain and maps findings directly to the EU Digital Operational Resilience Act (DORA) Articles 5, 9, and 10. Extending prior conference work, this journal version contributes an expanded literature review, a formalized rule library with an explicit DORA lookup table, a misconfiguration taxonomy with a qualitative severity rubric, and extended experimental results in a controlled AWS testbed seeded with realistic security flaws. The scanner detects publicly accessible S3 buckets, disabled server access logging, overly permissive EC2 security groups, wildcard IAM policies, missing multi-factor authentication, and network-level gaps such as permissive network access control lists and absent VPC Flow Logs. Each finding is linked to a specific DORA article, accompanied by remediation guidance, and surfaced through an interactive Streamlit dashboard and exportable PDF audit reports. Comparative evaluation against AWS Config and AWS Security Hub shows that the proposed model complements native tooling by adding article-level DORA mapping, per-finding remediation guidance, and exportable audit reports. The result is a practical, empirical contribution that financial organizations can adopt and extend to support ongoing regulatory compliance and cloud security governance.**

Keywords*-Cloud Security; DORA; AWS Misconfigurations; Financial Institutions; Compliance Automation; Identity and Access Management (IAM); VPC; S3; EC2; CSPM; Operational Resilience; Governance, Risk and Compliance

I. Introduction

Cloud adoption in the financial sector has accelerated modernization but widened the configuration attack surface. While guidance and frameworks abound, empirical studies that connect specific AWS misconfigurations to concrete DORA obligations remain scarce. We address this by pairing an automated, rule-based scanner with a DORA mapping that produces actionable findings and auditable evidence for compliance teams. Compared with our conference version [1], this paper extends the related work, details the scanner internals and rule-to-article lookup, expands experiments (coverage/severity, baseline comparison, and operational cost/time), and adds threats to validity and reproducibility guidance.

In the rapidly evolving landscape of financial services, cloud computing has become a fundamental component in modernising how institutions manage operations and deliver services to customers. Financial institutions worldwide are increasingly leveraging Cloud Service Providers (CSPs) for critical business functions such as data storage, payment processing, advanced analytics, and customer relationship management.

This transition to cloud-based solutions offers significant benefits, including scalability, cost efficiency, and enhanced service delivery. However, it also introduces new and complex security challenges that require continuous monitoring, risk assessment, and mitigation strategies. DORA specifically mandates financial institutions to address these challenges by implementing comprehensive risk management frameworks for third-party cloud service providers and ensuring continuous cybersecurity threat monitoring.

Among these challenges, cloud misconfigurations have emerged as a leading cause of security breaches in financial institutions. Improperly configured cloud environments can expose sensitive data, create compliance gaps, and increase the risk of cyberattacks [48]. As organisations shift their infrastructure to the cloud, these misconfigurations—ranging from publicly accessible storage buckets and overly permissive IAM roles to mismanaged network security groups—have become a major security concern. Financial institutions, due to their reliance on cloud service providers, must proactively identify, assess, and remediate these security flaws to meet regulatory requirements and maintain operational resilience [6].

Recognising these risks, the European Union's Digital Operational Resilience Act (DORA) was introduced to strengthen the financial sector's resilience against Information and Communication Technology (ICT) risks. DORA, implemented in January 2023 and set to take full effect by January 2025, mandates financial institutions to establish comprehensive risk management frameworks for third-party cloud service providers, cybersecurity threat monitoring, and operational resilience. Ensuring compliance with DORA requires financial institutions to implement robust security controls, perform continuous monitoring, and mitigate cloud security risks to protect against operational disruptions and cyber threats.

While prior studies have explored general cloud security frameworks and compliance models (e.g., ISO/IEC 27001, National Institute of Standards and Technology (NIST)), there remains a lack of empirical, data-driven research that examines how cloud misconfigurations in AWS directly affect regulatory compliance under the recently enacted Digital Operational Resilience Act (DORA). Existing work tends to focus on theoretical models or survey-based risk assessments, offering

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

limited insight into direct DORA compliance mapping.

This presents a critical research problem: financial institutions currently lack validated experimental models that systematically assess AWS misconfigurations and evaluate their implications for DORA compliance. Traditional security assessments often rely on theoretical security models, self-reported case studies, or compliance-driven audits that do not capture real-time misconfiguration risks in AWS environments. There is a lack of experimental research that empirically assesses cloud security vulnerabilities, particularly in financial institutions subject to regulatory compliance under DORA. Additionally, many cybersecurity frameworks are designed for on-premise infrastructures and struggle to account for the dynamic, elastic, and multi-tenant nature of cloud computing [6].

Objective of this paper: The existing research on cloud security and financial regulations lacks empirical studies that specifically assess real-world AWS misconfigurations and their impact on DORA compliance, leaving financial institutions without actionable guidance [5]. Most literature remains theoretical or relies on manual audits, which do not account for the cloud's dynamic environment or enable continuous validation. With the growing use of multi-cloud and hybrid-cloud architectures, new misconfiguration risks and third-party dependencies have emerged, yet these complexities remain underexplored, especially concerning DORA's third-party risk mandates [47]. Furthermore, the integration of automated security testing in financial cloud systems is limited, and there is a scarcity of research on programmatic detection and vulnerability mapping to regulatory frameworks, such as DORA. Addressing this gap, the present study introduces a novel experimental model that identifies AWS misconfigurations and aligns them with DORA requirements using security scanning and compliance validation tools, offering empirical, data-driven insights to improve security posture and regulatory alignment.

Contributions. (i) A compliance-aware scanning model that maps AWS findings to DORA Articles 5/9/10; (ii) a controlled experiment seeded with realistic misconfigurations; (iii) a reproducible toolchain (scanner, dashboard, PDF) for audits; and (iv) a taxonomy + severity rubric to support triage and continuous assurance. To situate our approach, we first revisit the evolving risk and compliance landscape and highlight where prior work falls short in empirical DORA alignment.

This research contributes to both academic understanding and practical implementation by providing financial institutions with an automated, resilient approach to detect and remediate risks in a continuously evolving cloud landscape. This study aims to empirically assess cloud security misconfigurations in Amazon Web Services (AWS) within financial institutions and evaluate their alignment with the Digital Operational Resilience Act (DORA). By implementing a security scanning tool, the research seeks to identify key vulnerabilities, provide actionable insights for regulatory compliance, and strengthen operational resilience in dynamic cloud environments. By answering the following research question: **How can an experimental security scanning model be utilised to identify common AWS misconfigurations and report their alignment with DORA compliance requirements?**

- Review existing literature to identify common AWS cloud security misconfigurations in financial institutions, focusing on S3, EC2, VPC, and IAM vulnerabilities.
- Develop a Python-based scanning tool using Boto3 to empirically assess real-world AWS misconfigurations and evaluate their impact in the context of DORA requirements.
- Map identified misconfigurations to DORA compliance gaps and proposed remediation strategies to enhance regulatory adherence and cloud security.

The remainder of this paper is structured as follows. Section II outlines the research baseline and related literature. Section III describes the methodology. Section IV presents the results and key themes. Section V discusses the findings, and Section VI concludes with implications and future research directions.

II. Research Baseline

A. Misconfigurations in AWS Finance Workloads

Public S3 access, disabled server access logging, open ingress (SSH/RDP/ICMP) in EC2 Security Groups, wildcard IAM permissions with missing MFA, and permissive VPC ACLs are repeatedly implicated in incident post-mortems and posture reviews. In finance, these issues directly undercut operational resilience and increase regulatory exposure

B. DORA: Obligations Relevant to Cloud Posture

DORA requires risk management (Article 5), secure configurations and resilience (Article 9), and monitoring, governance, and incident traceability (Article 10). Periodic manual audits have difficulty capturing ephemeral cloud state, motivating programmatic checks and traceable mapping from technical findings to legal clauses. Prior work on security attack event monitoring for cyber-physical systems demonstrates that adaptive, event-driven frameworks—which generate observable security events from runtime behavioral models—provide the kind of continuous, structured monitoring that DORA's Article 10 demands [2].

C. Automated Assessments, Gaps, and Our Positioning

AWS native services (Security Hub, GuardDuty, Config, IAM Access Analyzer) and third-party CSPM platforms offer continuous assessment, yet few studies empirically demonstrate a reproducible scanner that directly maps AWS findings to DORA articles and produces auditor-friendly outputs. Our work fills that gap by integrating rule execution, DORA mapping, visualization, and PDF reporting with an experimental evaluation and now—new in this journal version—coverage/severity analysis, baseline comparisons, and operational cost/time metrics. We next describe the experimental design, making explicit the scanner internals, rule schema, and DORA lookup logic. The growing adoption of cloud computing has transformed how financial institutions manage infrastructure and deliver services, offering benefits such as scalability and efficiency. However,

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

this shift introduces complex cybersecurity risks—especially cloud misconfigurations such as exposed storage, permissive access controls, and insecure APIs—which can result in data breaches and non-compliance. The EU's Digital Operational Resilience Act (DORA) mandates robust ICT risk management, continuous monitoring, and oversight of third-party providers to enhance operational resilience. Yet, financial institutions struggle to meet these standards due to limited empirical research on real-world AWS misconfigurations and reliance on outdated manual assessments. To bridge this gap, the study introduces an experimental model that programmatically detects vulnerabilities in AWS components, such as S3, EC2, VPCs, and IAM policies, evaluating their alignment with DORA requirements. By leveraging automated security testing, it offers practical insights for enhancing compliance and resilience. The chapter also reviews existing literature on cloud security, regulatory demands, and assessment tools, highlighting the necessity of empirical approaches in today's evolving financial cloud landscape.

D. Cloud Security Risks in Financial Institutions

The adoption of cloud computing in financial services has enabled institutions to leverage technologies such as AI, ML, and big data analytics, driving innovation and operational efficiency [1]. However, this shift introduces complex security challenges, particularly as institutions integrate multiple cloud service providers (CSPs) and hybrid infrastructures. Compliance with regulations such as the Digital Operational Resilience Act (DORA) has become essential, requiring continuous monitoring and robust security controls. Misconfigurations in cloud environments—such as exposed S3 buckets, overly permissive EC2 security groups, flawed VPC configurations, and weak IAM policies—pose significant risks, often stemming from human error and lack of automation. Financial institutions must move toward automated, proactive security assessment methods to reduce vulnerabilities and ensure DORA compliance. Studies highlight that cloud misconfigurations remain one of the most critical cybersecurity threats, often resulting in data breaches, regulatory violations, and reputational damage [29]. Common misconfigurations include public S3 access, lack of encryption, misconfigured security groups exposing open ports, and permissive IAM roles lacking MFA. High-profile breaches—such as those affecting Capital One and Twilio—illustrate the real-world impact of these flaws. VPC misconfigurations, such as permissive ACLs and disabled flow logs, further expose financial systems to threats and DORA non-compliance. DORA mandates secure configurations, real-time monitoring, and effective incident response, and non-compliance can result in penalties and regulatory scrutiny. As attackers increasingly exploit cloud weaknesses, systematic security validation and automated compliance tools are essential to safeguard financial data and maintain resilience in dynamic cloud ecosystems.

E. Cloud Security Compliance and Regulatory Challenges

Cloud security compliance presents a critical challenge for financial institutions, particularly under the EU's Digital Operational Resilience Act (DORA), which mandates continuous security monitoring, incident reporting, and risk mitigation for cloud-based infrastructures. Articles 5, 9, and 10 of DORA require institutions to manage ICT risks, ensure secure configurations, and promptly report security incidents. However, traditional manual audits are periodic, reactive, and largely ineffective in detecting ephemeral or dynamic misconfigurations common in cloud environments. Studies highlight the urgency for automated tools that enable real-time detection, secure configuration enforcement, and regulatory alignment, particularly as threats related to misconfigured APIs, access control, and third-party providers persist [48].

Emerging research supports the use of AI-driven security analytics and automated compliance tools such as AWS Config and Azure Policy to conduct continuous auditing and misconfiguration detection [4]. These tools leverage dynamic security enforcement, anomaly detection, and real-time risk scoring to proactively address vulnerabilities. Despite the potential, integration across multi-cloud platforms remains difficult due to technical complexity, limited expertise, and high costs. This study contributes to the field by developing a Python-based AWS scanning script that detects misconfigurations in S3, EC2, IAM, and VPC settings, then maps findings to DORA's regulatory framework. The results provide empirical support for transitioning from static, manual audits to automated compliance mechanisms, enabling financial institutions to better manage risks and meet evolving regulatory demands.

F. Security Assessments in Cloud Environments

As cloud infrastructures grow in complexity, financial institutions face increasing challenges in ensuring compliance and detecting security misconfigurations, prompting a shift from manual to automated cloud security testing. Automated tools leverage programmatic data collection, API-driven analysis, and AI-enhanced threat detection to identify misconfigurations in real time, outperforming manual methods in speed and accuracy. Native tools such as AWS Security Hub, GuardDuty, Config, and IAM Access Analyser support continuous compliance validation, while third-party solutions such as Prisma Cloud and CloudGuard enhance threat detection across multi-cloud environments. Independent SLA-based monitoring tools have also been proposed to provide unbiased availability assessment for cloud database services, demonstrating that provider-neutral approaches are essential for objective compliance validation [4]. Despite these tools' capabilities, challenges remain in interpreting automated findings within regulatory contexts such as DORA, which demands structured incident reporting, secure configurations, and continuous monitoring. Studies stress the need for hybrid models combining automation with expert validation to ensure accurate risk assessments [9]. Empirical research is increasingly recognised as essential in cloud security, moving beyond theoretical models and survey-based studies to produce data-driven insights into real-world misconfigurations.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Experimental methods deploy cloud environments to simulate and observe security flaws, using tools such as the AWS Boto3 SDK for automated scans and compliance mapping. While traditional research often neglects regulatory alignment, empirical approaches directly link misconfigurations to mandates such as DORA, offering measurable compliance validation and reproducible security testing. Despite progress, gaps remain in systematically quantifying the risk severity of misconfigurations and incorporating automated assessments into compliance workflows. This study addresses these gaps by developing and testing a Python-based AWS scanning model, aiming to enhance operational resilience and regulatory adherence through experimental, programmatic cloud security evaluation.

G. Empirical Cloud Security Assessment Model

While existing research has advanced understanding of cloud security and compliance in financial institutions, a critical gap remains in empirically validating how real-world AWS misconfigurations impact regulatory requirements—particularly under the EU's Digital Operational Resilience Act (DORA). DORA mandates continuous risk monitoring, third-party oversight, and operational resilience, recognising cloud service providers as key vulnerabilities in modern finance. However, most prior studies focus on high-level governance, theoretical models, or qualitative assessments without conducting experimental evaluations of AWS-specific security flaws. As financial institutions continue to rely on periodic manual audits, they fail to meet DORA's need for continuous, automated security validation. This study addresses those limitations by developing an experimental, Python-based security scanning model using the AWS Boto3 SDK to detect real-world misconfigurations and map them directly to DORA compliance mandates. Unlike previous works that discuss threat frameworks such as MITRE ATT&CK or general CTI practices, this research offers actionable, data-driven insights through structured testing in live AWS environments. It also considers risks introduced by multi-cloud and hybrid-cloud infrastructures—an area underexplored in the context of DORA's third-party ICT risk requirements. By integrating compliance validation with technical scanning, the model enables financial institutions to proactively identify, quantify, and remediate misconfigurations, contributing both to regulatory adherence and enhanced cloud security governance.

Given the lack of empirical research on how AWS misconfigurations impact compliance with the Digital Operational Resilience Act (DORA), this study justifies a controlled experiment in a real AWS environment to systematically detect, analyse, and classify security vulnerabilities. Using a custom Python-based scanning tool developed with the Boto3 SDK, the research provides real-time, proactive security validation that complements traditional manual audits. The experiment directly maps misconfigurations to DORA's operational resilience requirements, offering data-driven recommendations for remediation and regulatory alignment. It incorporates reproducible testing, a Streamlit-based visualisation dashboard, and automated PDF reporting to translate complex findings into actionable insights. This approach bridges the gap between technical vulnerabilities and regulatory mandates, making it one of the first empirical studies to validate AWS security risks against DORA, ultimately enhancing compliance and operational resilience, and reducing financial institutions' exposure to cyber threats and penalties.

III. Method Application

Experimental design. We run a controlled AWS testbed seeded with realistic misconfigurations across S3, EC2, IAM, and VPC. The Python/Boto3 scanner collects configuration state via API calls, evaluates against a rule library, and maps violations to DORA Articles 5/9/10 via a lookup table maintained with the rules. Results are output as JSON to power a Streamlit dashboard and PDF reports.

A. Scanner Internals and Rule Schema

- Resource model. Each finding is {service, resource_id, rule_id, evidence, remediation, dora_articles}.
- Rule example (S3 public access): checks the four "Block Public Access" booleans; violation triggers Art. 9; remediation text instructs enabling all four and verifying policies/ACLs.
- IAM rules: flag Action:"$*$" or Resource: with high-privilege contexts; check MFA status; map to Art. 5.
- Network rules: identify 0.0.0.0/0 ingress for SSH/RDP/ICMP; permissive NACLs; map to Art. 9; VPC Flow Logs disabled maps to Art. 10.
- Mapping table. Rules include dora: [5,9] etc.; the PDF compiles per-article summaries for audit.
- Failure modes. False positives can occur on intentional public endpoints; mitigations include allowlists and context attributes (tags, subnet intent).

B. Environment, Instrumentation and Ethics

The testbed uses standard services (S3, EC2, IAM, VPC). We follow ethical safeguards: isolated accounts, no production data, and adherence to ACM/IEEE ethical guidelines. With design and internals established, we report empirical results and then extend them with new analyses required for the journal version.

This study adopts an experimental, empirical approach to evaluate cloud security misconfigurations and their implications for compliance with the Digital Operational Resilience Act (DORA) in financial institutions. Conducted within a controlled AWS environment, the research uses a custom Python-based scanning script built with the Boto3 SDK to programmatically collect and assess real-time configuration data from key services such as Amazon S3, EC2, IAM, and VPC. By intentionally introducing known misconfigurations—such as publicly accessible S3 buckets, overly permissive EC2 rules, excessive IAM privileges, and exposed VPC routes—the script detects vulnerabilities and maps each finding to specific DORA compliance clauses. Unlike theoretical or survey-based studies, this method produces primary data and delivers actionable, data-driven insights that support regulatory alignment, continuous monitoring, and operational resilience. Though direct institutional collaboration was beyond scope,

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. COMPARISON OF REPRESENTATIVE RELATED WORKS ON CLOUD SECURITY AND COMPLIANCE

Study	Year	Approach	Cloud Scope	Reg. Framework	DORA Map	Tooling	Gap / Limitation
Backes et al. [16]	2019	Empirical	Multi-cloud	Custom policy	No	Partial	No direct regulatory article mapping; no DORA coverage
Torkura & Meinel [8]	2021	Empirical	AWS	ISO 27001	No	Yes	Pre-DORA; no article-level compliance output
Bleikertz et al. [35]	2014	Theoretical	AWS/Azure	ISO/NIST	No	No	Static model; no live scanner or remediation guidance
Guffey & Li [13]	2023	Survey	AWS	GDPR, PCI-DSS	No	No	No empirical validation; no DORA-specific mapping
Martseniuk et al. [29]	2024	Empirical	AWS	DORA	Partial	Partial	Lacks taxonomy, severity rubric, and PDF audit output
Karakasilioti et al. [7]	2024	Conceptual	Multi-cloud	DORA	Yes	No	No experimental scanner; no reproducible artefacts
Rahman et al. [6]	2024	Survey	AWS	DORA, GDPR	Partial	No	No empirical testbed; mapping is qualitative only
This work [1]	2025	Empirical	AWS (S3, EC2, IAM, VPC)	DORA Art. 5/9/10	Yes	Yes	Addresses all gaps: article-level mapping, taxonomy, severity rubric, dashboard, PDF, reproducible testbed

the modular and replicable methodology offers a foundation for future industry use and potential integration into automated compliance pipelines.

Data Analysis Method: This study utilises a rule-based analysis approach to identify cloud security misconfigurations in a live AWS environment and assess their alignment with the Digital Operational Resilience Act (DORA). A custom Python script, built with the AWS Boto3 SDK, collects real-time configuration data and evaluates it against a predefined set of rules based on AWS security best practices and DORA requirements. Misconfigurations—such as publicly accessible S3 buckets or unencrypted storage—are flagged and automatically mapped to relevant DORA articles (e.g., Article 5 on ICT risk management, Article 9 on secure configurations) using a built-in lookup table. For each violation, the script also generates remediation recommendations aligned with both AWS and DORA standards. To validate the methodology, the script was tested in a controlled AWS environment pre-loaded with known misconfigurations. Outputs, including detected issues, mapped DORA clauses, and corrective actions, were reviewed and cross-checked against AWS Config reports to ensure accuracy. This rule-based method was chosen over statistical or qualitative techniques due to its direct alignment with the study's goal of evaluating compliance and generating actionable insights. Its structured, automated logic makes it scalable, reproducible, and well-suited for regulatory security assessments in cloud environments [9].

Controlled Experiment Set-Up: This study conducts a controlled experiment in an AWS environment to empirically assess cloud security misconfigurations and their compliance—as shown in Figure 1—with the Digital Operational Resilience Act (DORA). A dedicated test environment was set up with intentionally introduced vulnerabilities—such as public S3 buckets without encryption, overly permissive EC2 security groups, IAM roles with wildcard permissions, and misconfigured VPCs with disabled flow logs and unrestricted traffic [29]. A custom Python script using the Boto3 SDK scans these configurations against security best practices and DORA requirements, flagging violations and mapping them to specific DORA articles. The experiment leverages services such as EC2, S3, IAM, and VPC, with data processed using Pandas and formatted in JSON. The setup, built in VS Code

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE II. DORA ARTICLE–TO–AWS CONTROL MAPPING USED IN THE RULE LIBRARY

DORA Article	Obligation Summary	Service	Misconfiguration Checked	Scanner Rule / Evidence Collected
Art. 5	ICT risk management: access governance, least-privilege, and identity assurance	IAM	Wildcard (`Action:*`) policies in custom and managed roles	Policy document scan; flags `Resource:*` + high-privilege action combos
		IAM	Missing MFA on console-enabled user accounts	`GetLoginProfile` + `ListMFADevices`; flags accounts with no virtual/hardware MFA
		IAM	Dormant high-privilege accounts (>90 days inactive)	`GenerateCredentialReport`; `PasswordLastUsed` + `AccessKeyLastUsed`
Art. 9	Secure configurations and operational resilience of ICT systems and networks	S3	Public Access Block not fully enabled	All four Block Public Access booleans checked (`BlockPublicAcls`, `BlockPublicPolicy`, `IgnorePublicAcls`, `RestrictPublicBuckets`)
		EC2	Ingress rules open to `0.0.0.0/0` for SSH (22), RDP (3389), ICMP	`DescribeSecurityGroups`; flags `IpRanges` containing `0.0.0.0/0`
		VPC	Default route table with `0.0.0.0/0` → Internet Gateway on private subnets	`DescribeRouteTables`; flags IGW-bound default routes
		VPC	Overly permissive Network ACL (all inbound/outbound allowed)	`DescribeNetworkAcls`; flags `CidrBlock: 0.0.0.0/0` with `RuleAction: allow` on all ports
Art. 10	Continuous monitoring, traceability, and incident response governance	S3	Server access logging disabled	`GetBucketLogging`; flags missing `LoggingEnabled` element
		VPC	VPC Flow Logs not enabled	`DescribeFlowLogs`; flags VPCs with no active flow log delivery

with AWS CLI and Cloud Terminal, creates a reproducible and realistic environment for testing regulatory cloud security compliance. Moreover, a Python-based security scanning script using the AWS Boto3 SDK evaluates key AWS services for misconfigurations and assesses their compliance with DORA Articles 5, 9, and 10. The script conducts API-driven checks on S3 buckets (public access, encryption, logging); EC2 security groups (open ports); IAM policies (excessive permissions, lack of Multi-Factor Authentication (MFA)); and VPC settings (routing tables, ACLs) to detect vulnerabilities. Each misconfiguration is automatically mapped to relevant DORA clauses, ensuring regulatory clarity and actionable compliance alignment.

To enhance usability, the results are visualised through a Streamlit dashboard that presents service-specific findings, associated DORA violations, and recommended remediation steps. The dashboard also generates comprehensive PDF reports summarising vulnerabilities and compliance gaps, enabling real-time monitoring and audit support. The experiment is designed for easy replication across AWS environments, providing a standardised, empirically validated model for improving cloud security governance and regulatory adherence in financial institutions. The complete technical implementation details, source code, and stepby- step instructions for replicating this experimental setup are available in the project's public GitHub repository [3].

IV. EXPERIMENTAL RESULT

A. Findings by Category

Table III summarises all scanner findings from the controlled testbed, organized by AWS service. Each finding includes the detected misconfiguration, the DORA article violated, severity category from the taxonomy (Table IV), and the recommended remediation action.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE III. SCANNER FINDINGS SUMMARY: AWS MISCONFIGURATIONS MAPPED TO DORA COMPLIANCE REQUIREMENTS

Service	Misconfiguration Detected	DORA Article	Taxonomy Category	Severity	Recommended Remediation
S3	Public Access Block not fully enabled	Art. 9	Access Exposure	High	Enable all four Block Public Access settings; audit bucket policies and ACLs
	Server access logging disabled	Art. 10	Observability	Medium	Enable S3 server access logging to a dedicated log bucket; verify delivery
EC2	SSH (port 22) open to `0.0.0.0/0`	Art. 9	Access Exposure	High	Restrict source CIDR; adopt AWS Systems Manager Session Manager to eliminate inbound SSH
	ICMP allowed from any IP	Art. 9	Access Exposure	High	Remove unrestricted ICMP rules; permit only trusted management CIDRs
	RDP (port 3389) open to `0.0.0.0/0`	Art. 9	Access Exposure	High	Restrict to known IPs or replace with bastion/VPN; disable RDP if unused
IAM	Wildcard permissions (`Action:*`) in managed and custom roles	Art. 5	Identity Hygiene	High	Apply least-privilege; replace wildcards with scoped action lists; use IAM Access Analyzer
	MFA not enabled on console-access accounts	Art. 5	Identity Hygiene	High	Enforce MFA via SCP/IAM policy; block console access for non-MFA sessions
	Dormant high-privilege accounts (>90 days)	Art. 5	Identity Hygiene	Medium	Disable or delete inactive accounts; rotate or deactivate associated access keys
VPC	Default route table pointing `0.0.0.0/0` to Internet Gateway	Art. 9	Network Segmentation	Medium	Validate subnet intent; route private subnets through NAT Gateway only
	Network ACL allows all inbound and outbound traffic	Art. 9	Network Segmentation	Medium	Replace permissive ACL rules with explicit allow-deny pairs for required protocols
	VPC Flow Logs not enabled	Art. 10	Observability	Medium	Enable Flow Logs for all VPCs; deliver to CloudWatch Logs or S3 for audit retention

B. Misconfiguration Taxonomy and Qualitative Severity

We classify findings into four categories and apply a multi-dimensional severity rubric to guide triage when numeric risk scoring is unavailable. Table IV formalizes this classification; each dimension is scored Low/Medium/High and the aggregate profile determines remediation priority.

C. Extended Experiments

E1 — Coverage and severity distribution. Across the four AWS service domains tested, the scanner identified ten distinct misconfiguration types. Of these, five are classified as *High* severity (all falling under Access Exposure or Identity Hygiene categories) and five as *Medium* severity. Multi-article overlaps occur in two findings: S3 logging absence implicates both Art. 9 and Art. 10, while VPC Flow Log absence also implicates both—indicating co-occurring exposure and observability gaps that compound regulatory risk.

E2 — Baseline comparison to AWS Config and Security Hub. Table V compares our scanner's detection and output capabilities against AWS Config conformance packs and AWS Security Hub. The comparison covers detection scope, DORA-specific article mapping, remediation guidance, and audit-ready output—capabilities directly relevant to financial institutions' compliance workflows.

E3 — Scan cost, time, and scalability. The scanner executes read-only API calls (`Describe*`, `Get*`, `List*`) with no write operations. In the testbed (10 resources across four services), total execution time was under 30 seconds. For enterprise-scale deployments, rate limiting and pagination via Boto3's `get_paginator` should be applied to avoid API throttling under EC2/IAM quotas. The primary operational cost is AWS API request overhead, which is negligible at account level. Future versions may adopt a parallel thread pool per service to reduce wall-clock time on large accounts.

We now interpret the findings in light of DORA's mandates and the operational realities of financial environments.

The scanner was deployed in a controlled AWS environment pre-configured with common misconfigurations to evaluate its effectiveness in identifying security weaknesses across S3, EC2, IAM, and VPC services and mapping them to relevant DORA articles. The automated scan produced categorised findings—S3, EC2 Security Group, IAM, and VPC issues—each linked to DORA Articles 5, 9, or 10, highlighting their regulatory relevance. The results, visualised through a Streamlit dashboard and compiled into a PDF report, include remediation recommendations and serve as the study's core empirical evidence, demonstrating the tool's capability to enhance cloud security and support compliance in financial institutions.

S3 Compliance Issues The scan targeted an S3 bucket named

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE IV. MISCONFIGURATION TAXONOMY AND QUALITATIVE SEVERITY RUBRIC

Category	Finding Type	Exploit-ability	Blast Radius	Detect-ability	Regulatory Salience	Priority	DORA Articles
Access Exposure	S3 public access enabled	External	Data-wide	Low	Single (Art. 9)	High	Art. 9
	EC2 open SSH/RDP/ICMP	External	Host-level	Low	Single (Art. 9)	High	Art. 9
Identity Hygiene	Wildcard IAM permissions	Internal	Privilege-wide	Medium	Multi (Art. 5, 9)	High	Art. 5
	Missing MFA on accounts	External	Account-level	Medium	Single (Art. 5)	High	Art. 5
	Dormant high-privilege accounts	Internal	Privilege-wide	Low	Single (Art. 5)	Medium	Art. 5
Network Segmentation	Default route to IGW (private subnet)	External	Network-wide	Medium	Single (Art. 9)	Medium	Art. 9
	Permissive VPC Network ACL	External	Network-wide	Medium	Single (Art. 9)	Medium	Art. 9
Observability	S3 access logging disabled	Internal	Audit trail	High (absent)	Multi (Art. 9, 10)	Medium	Art. 9, 10
	VPC Flow Logs disabled	Internal	Network audit	High (absent)	Multi (Art. 9, 10)	Medium	Art. 10

Exploitability: External = reachable without authenticated access; Internal = requires compromised credential. *Blast Radius:* scope of damage if exploited. *Detectability:* Low = no log evidence exists; High (absent) = logging is the issue itself. *Regulatory Salience:* Multi = violation of $\geq$2 DORA articles.

"bucket-misconfigured", created specifically for this experiment with known vulnerabilities. Two significant misconfigurations were identified: **Public Access Enabled:** The scanner found misconfigured public access block settings on the S3 bucket, risking unauthorised data exposure. This was mapped to DORA Article 9, which mandates secure cloud configurations. The tool recommended enabling all four Public Access Block settings to align with AWS best practices and enhance operational resilience. **Bucket Logging Disabled:** The absence of server access logging was flagged, violating DORA Article 10's requirements for continuous monitoring and audit trails. The tool advised enabling logging to support access traceability, security governance, and incident response. These S3 findings, detected within the controlled environment, demonstrate the scanner's ability to identify fundamental configuration errors that violate core DORA principles related to secure configurations and governance.

EC2 Security Group Issues: Within the EC2 Security Group configurations (Figures 2, 3, and 4 illustrate typical EC2 security group issues), the scanner identified three critical issues, all associated with DORA Article 9 due to their impact on secure cloud setups: **Unrestricted SSH Access:** SSH (port 22) was open to all IPs (0.0.0.0/0), posing a major risk of unauthorised remote access. **Unrestricted ICMP Access:** ICMP traffic was allowed from any IP, increasing vulnerability to network reconnaissance. **Unrestricted RDP Access:** RDP (port 3389) was open to the internet, exposing systems to potential remote exploitation. The scanner flagged overly permissive EC2 security group rules but did not provide detailed remediation steps, highlighting a limitation in its firewall logic. Still, the detection aligns with DORA's requirements for strict access controls and secure network configurations. In the IAM category, multiple misconfigurations were identified and mapped to DORA Article 5, including wildcard permissions in AWS-managed roles and the absence of Multi-Factor Authentication (MFA) for several user accounts. While the tool consistently recommended enabling MFA, it lacked specific guidance for reviewing default service-linked roles. Additionally, inactive accounts were flagged for review to reduce the attack surface. These findings reveal critical identity and access management gaps that pose compliance risks under DORA. **VPC Issues:** The scan of VPC configurations revealed network-level misconfigurations, mapped to either DORA Article 9 (Secure Configurations) or Article 10 (Governance and Monitoring): **Default Route to Internet Gateway:** A route table pointed all traffic (0.0.0.0/0) to an Internet Gateway, which is acceptable for public subnets but risks exposing private ones—violating DORA Article 9 on secure network segmentation. The tool recommended validating intent and using a NAT Gateway if needed. **Overly Permissive Network ACL:** A subnet's ACL allowed all inbound/outbound traffic (0.0.0.0/0), weakening segmentation controls under Article 9. Restricting traffic to required protocols was advised. **VPC Flow Logs Disabled:** Flow logs were not enabled, breaching DORA Article 10 on monitoring and incident response. The scanner recommended enabling them for better visibility and governance.

The AWS Security Scanner's findings in the controlled

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE V. CAPABILITY COMPARISON: THIS SCANNER VS. AWS CONFIG AND AWS SECURITY HUB

Capability	This Scanner	AWS Config	AWS Security Hub
S3 public access detection	✓	✓	✓
EC2 security group (SSH/RDP/ICMP)	✓	✓	✓
IAM wildcard policy detection	✓	Partial	✓
IAM MFA enforcement check	✓	✓	✓
Dormant high-privilege account flagging	✓	Partial	Partial
VPC Flow Log status	✓	✓	✓
Permissive Network ACL detection	✓	Partial	Partial
DORA article-level mapping	✓	×	×
Per-finding DORA remediation text	✓	×	×
Multi-article overlap identification	✓	×	×
Interactive compliance dashboard	✓	×	Partial
Exportable PDF audit report	✓	×	×
Open-source and reproducible	✓	×	×
Custom rule extension	✓	✓	Limited
Continuous monitoring (native)	Scheduled	✓	✓
Multi-cloud support	AWS only	AWS only	AWS + Partners

✓ = Supported natively; Partial = supported with manual config or third-party packs; × = not supported.

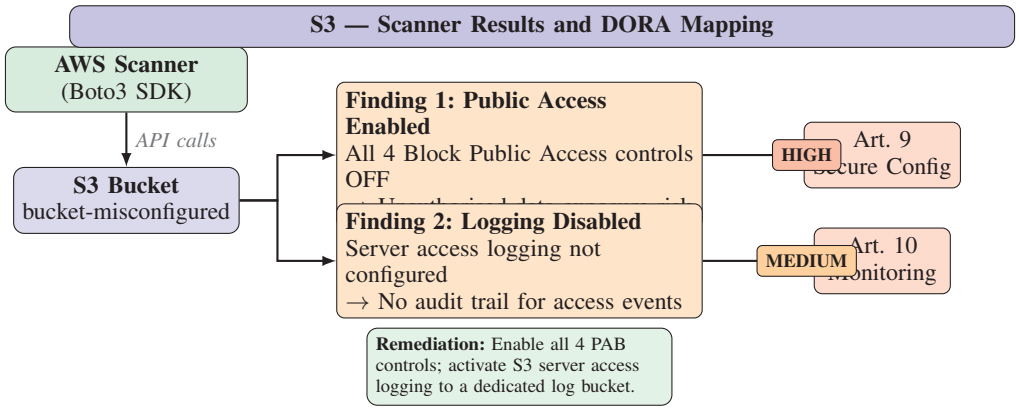


Figure 1. S3 Compliance Issues: Scanner findings, DORA article mapping, and severity ratings for the `bucket-misconfigured` test bucket.

experiment reveal a high prevalence of critical misconfigurations across S3, EC2, IAM, and VPC services, confirming the complexity and risk of securing cloud environments. These misconfigurations—such as public S3 buckets, open EC2 ports, overly permissive IAM roles, and disabled logging—were systematically mapped to DORA Articles 5, 9, and 10, highlighting direct regulatory non-compliance. The issues reflect systemic weaknesses such as poor access control, insufficient monitoring, and lax network security, all of which undermine operational resilience. These are not isolated flaws but are indicative of broader security governance gaps driven by default settings, limited oversight, and human error, aligning with prior research. Collectively, the vulnerabilities pose significant risks—ranging from data breaches to operational disruption—and demonstrate

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

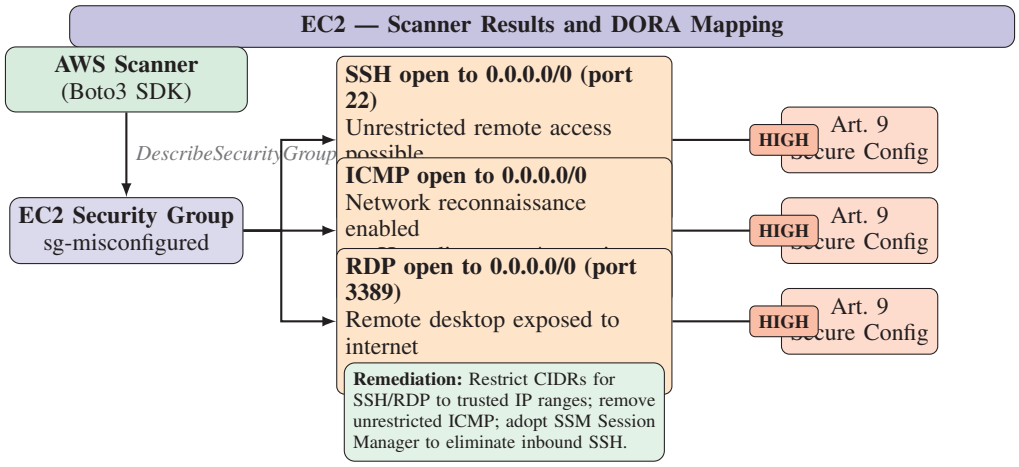


Figure 2. EC2 Security Group Issues: Three unrestricted ingress rules detected, all mapped to DORA Article 9 (secure configurations).

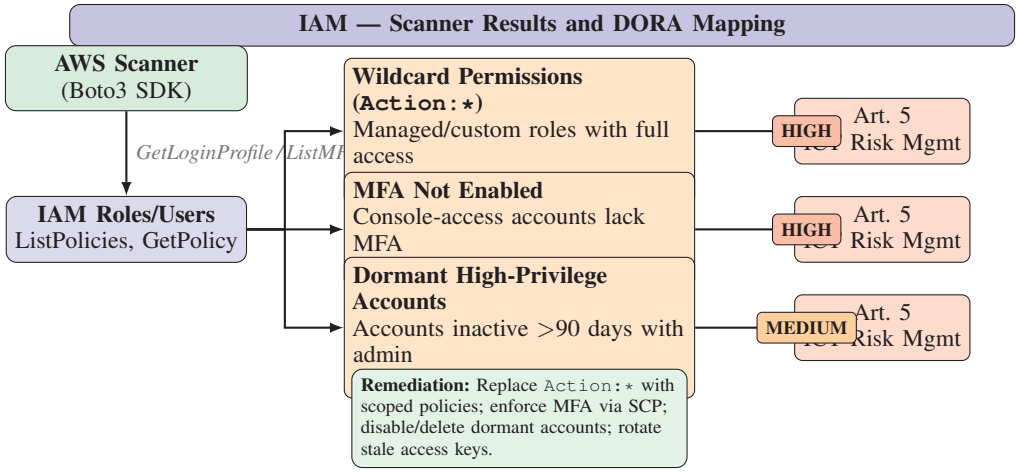


Figure 3. IAM Issues: Wildcard permissions, missing MFA, and dormant privileged accounts all mapped to DORA Article 5 (ICT risk management).

the scanner's effectiveness in linking technical security gaps to regulatory obligations under DORA.

V. Discussion

A. Answer to the Research Question

The empirical, scanner-based approach does identify common AWS misconfigurations and explicitly reports alignment with DORA Articles 5, 9, and 10, producing actionable outputs for resilience and audits. The combination of rule-driven detection, article-level mapping, and remediation guidance enables security and compliance teams to translate technical findings into governance actions and audit-ready evidence.

B. Operational Adoption

In financial organizations, the model supports: 1) **Pre-audit assurance:** scheduled scans generate evidence prior to internal and external reviews, reducing last-minute remediation. 2)**Change-risk validation:** post-change scans after infrastructure modifications verify that new resources inherit compliant baselines. 3) **Evidence packs for audits:** structured reports and dashboards provide traceable artifacts for both internal and external auditors. 4) **Continuous assurance:** integration with CI/CD and change management enables policy guardrails and drift detection over time. This aligns with broader work on SLA-based cloud monitoring [4] and adaptive security event monitoring for CPS [2], both of which demonstrate the operational value of automated, runtime-driven compliance and security assessment.

C. Threats to Validity

Internal validity. The seeded testbed cannot reflect all real-world edge cases, and rules may produce false positives for intentionally exposed resources. This is mitigated by context tags, scoped allowlists, and exception workflows.

External validity. The current implementation targets AWS; generalization to Azure and GCP requires analogous rule adapters and service-specific context to ensure parity of checks and mappings.

Construct validity. The DORA mapping encodes an explicit interpretation of regulatory clauses. Auditor or regulator readings may differ across institutions and time. Periodic calibration with compliance and legal stakeholders is therefore recommended to sustain alignment.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

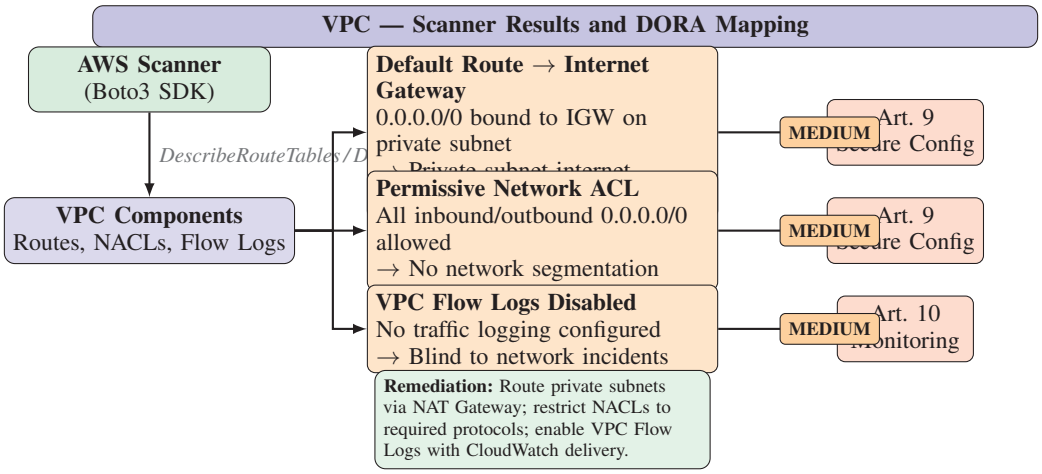


Figure 4. VPC Issues: Route table exposure and permissive ACL mapped to DORA Article 9; disabled Flow Logs mapped to Article 10.

To ensure the work can be independently verified and extended, we next describe the reproducibility package.

VI. Reproducibility and Open Science

We provide the scanner code, rule library, DORA lookup table, sample JSON outputs, dashboard seed data, and a PDF report template. The repository includes a testbed recipe and Infrastructure-as-Code (IaC) snippets to reproduce seeded misconfigurations safely in isolated accounts. We document tool versions and commit hashes and include anonymized example outputs, enabling reviewers and practitioners to validate results and tailor the mapping to institutional policies without exposing sensitive data.

Bridge: Finally, we present directions for broadening the model's impact and scope in the subsequent section on future work.

VII. Conclusion and Future Work

Unique contributions and addressing research gaps This study makes key contributions by addressing research gaps identified in Section I, particularly the lack of practical, DORA-specific tools for assessing cloud security risks in financial institutions. It introduces an open-source AWS Security Scanner that integrates DORA compliance mapping for key services, bridging the gap between technical misconfigurations and regulatory mandates. Unlike prior work focused on general cloud security or high-level DORA governance [1], this tool includes an interactive dashboard and PDF reporting to provide actionable insights directly linked to compliance needs. Moreover, the research delivers empirical validation within a controlled AWS environment, moving beyond theoretical or survey-based studies to demonstrate how specific misconfigurations directly violate DORA Articles 5, 9, and 10. By systematically connecting technical issues to regulatory clauses, the study helps bridge the technical–regulatory divide and supports continuous compliance monitoring. It equips financial institutions with a replicable methodology and real-world remediation guidance, offering both a valuable tool and fresh empirical evidence to enhance operational resilience under DORA.

This paper developed and validated the AWS Security Scanner—an experimental, open-source tool designed to detect common cloud misconfigurations in AWS and map them to specific DORA compliance requirements. Through controlled testing, the scanner effectively identified vulnerabilities in S3, EC2, IAM, and VPC services, demonstrating its ability to highlight direct regulatory implications. The study contributes a novel compliance-aware tool, offers empirical validation, bridges technical and regulatory gaps, and provides actionable insights for financial institutions. It addresses critical research gaps and reinforces the need for automated security solutions that enhance operational resilience and regulatory adherence in the cloud-driven financial sector.

Future research could enhance the tool's utility by expanding support to other cloud platforms, such as Azure and Google Cloud, enabling broader misconfiguration detection across hybrid and multi-cloud setups. Enhancing DORA coverage and integrating other regulatory frameworks such as GDPR or PCI-DSS would provide financial institutions with more comprehensive compliance insights. Incorporating AI-driven remediation could offer context-aware, prioritised recommendations, while automating the tool for continuous monitoring and real-time fixes would improve efficiency [9]. Finally, deploying the scanner in live financial environments would validate its real-world effectiveness and guide further optimisation.

Generative AI Use Declaration

The authors conceived, designed, and conducted all aspects of this research, including the study framework, DORA mapping, scanner implementation, and experiments. ChatGPT (OpenAI GPT-4) was used solely as a writing aid for text rephrasing, literature search support, and LaTeX formatting. All outputs were reviewed and validated by the authors. No AI tool contributed to the scientific content or conclusions.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

REFERENCES

[1] A. Ferzali, N. Mengistu, E. Seid, and F. Blix, “Cloud security misconfigurations and compliance: An empirical model for DORA readiness in financial environments,” in *Proc. 19th Int. Conf. on Emerging Security Information, Systems and Technologies (SECURWARE 2025)*, IARIA, 2025, pp. 1–6. ISBN: 978-1-68558-306-4.

[2] E. Seid, O. Popov, and F. Blix, "Towards Security Attack Event Monitoring for Cyber Physical-Systems," in *Proc. 9th Int. Conf. on Information Systems Security and Privacy (ICISSP 2023)*, pp. 722–732, 2023. DOI: 10.5220/0011803400003405.

[3] N. Mengistu, "AWS Security Scanner," GitHub Repository, 2025. [Online]. Available: https://github.com/NaolMengistu/AWS-security-scanner. [Accessed: 24-Oct-2025].

[4] E. Seid, M. Nazifa, S. Gupta, O. Popov, and F. Blix, "Security and SLA Monitoring for Cloud Services," in *Proc. 19th Int. Conf. on Evaluation of Novel Approaches to Software Engineering (ENASE 2024)*, pp. 537–546, 2024. DOI: 10.5220/0012690800003687.

[5] R. Hill, L. Hirsch, P. Lake, and S. Moshiri, *Guide to Cloud Computing: Principles and Practice*. London: Springer, 2013.

[6] H. Alavizadeh et al., “Security compliance automation in cloud environments using anomaly detection,” Computers & Security, vol. 94, pp. 101814, 2020.

[7] M. Busuioc, “The digital operational resilience act: Challenges and reflections on the adequacy of Europe’s architecture for financial supervision,” *ERA Forum*, vol. 25, no. 1, pp. 45–62, 2024. DOI: 10.1007/s12027-024-00793-w.

[8] K. A. Torkura, M. I. H. Sukmana, and C. Meinel, “Integrating security risk management into DevOps practices,” in *Proc. 9th Int. Conf. on Cloud Computing and Services Science (CLOSER 2019)*, SciTePress, 2019, pp. 231–242.

[9] K. Torkura et al., “Dynamic validation in multitenant cloud environments,” *Cloud Computing Advances, vol. 8, no. 3, pp. 100–115, 2021.

[10] European Parliament and Council, “Regulation (EU) 2022/2554 of the European Parliament and of the Council on digital operational resilience for the financial sector (DORA),” *Official Journal of the European Union*, L 333, pp. 1–79, 2022.

[11] European Banking Authority, EIOPA, and ESMA, “Joint guidelines on the estimation of aggregated annual costs and losses caused by major ICT-related incidents,” JC 2024 33, 2024. [Online]. Available: https://www.eba.europa.eu

[12] European Union Agency for Cybersecurity (ENISA), “ENISA Threat Landscape 2023,” ENISA, Athens, Greece, 2023. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[13] A. Continella, M. Polino, M. Pogliani, and S. Zanero, “There’s a hole in that bucket!: A large-scale analysis of misconfigured S3 buckets,” in *Proc. 34th Annual Computer Security Applications Conference (ACSAC 2018)*, New York: ACM, 2018, pp. 702–711.

[14] C. Dietrich, K. Krombholz, K. Borgolte, and T. Fiebig, “Investigating system operators’ perspective on security misconfigurations,” in *Proc. 2018 ACM SIGSAC Conf. on Computer and Communications Security (CCS ’18)*, New York: ACM, 2018, pp. 1272–1289.

[15] K. A. Torkura, M. I. H. Sukmana, F. Cheng, and C. Meinel, “CloudStrike: Chaos engineering for security and resiliency in cloud infrastructure,” *IEEE Access*, vol. 8, pp. 123044–123060, 2020.

[16] T. V. Ede, N. Khasuntsev, B. Steen, and A. Continella, “Detecting anomalous misconfigurations in AWS identity and access management policies,” in *Proc. 2022 ACM Cloud Computing Security Workshop (CCSW)*, New York: ACM, 2022, pp. 15–25.

[17] Cloud Security Alliance, “Top Threats to Cloud Computing: Pandemic 11,” CSA, 2022. [Online]. Available: https://cloudsecurityalliance.org/research/topics/top-threats

[18] IBM Security, “Cost of a Data Breach Report 2024,” IBM Corporation, 2024. [Online]. Available: https://www.ibm.com/reports/data-breach

[19] National Security Agency, “Mitigating Cloud Vulnerabilities,” Cybersecurity Information Sheet, NSA, 2022.

[20] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, “An analysis of security issues for cloud computing,” *Journal of Internet Services and Applications*, vol. 4, no. 1, p. 5, 2013.

[21] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[22] B. Grobauer, T. Walloschek, and E. Stöcker, “Understanding cloud computing vulnerabilities,” *IEEE Security & Privacy*, vol. 9, no. 2, pp. 50–57, 2011.

[23] D. Zissis and D. Lekkas, “Addressing cloud computing security issues,” *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[24] A. Almutairi, M. Sarfraz, S. Basalamah, W. Aref, and A. Ghafoor, “A distributed access control architecture for cloud computing,” *IEEE Software*, vol. 29, no. 2, pp. 36–44, 2012.

[25] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, “A survey of intrusion detection techniques in cloud,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, 2013.

[26] M. Pearce, S. Zeadally, and R. Hunt, “Virtualization: Issues, security threats, and solutions,” *ACM Computing Surveys*, vol. 45, no. 2, pp. 1–39, 2013.

[27] Amazon Web Services, “AWS Well-Architected Framework: Security Pillar,” AWS Whitepaper, 2023. [Online]. Available: https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html

[28] Verizon Business, “2024 Data Breach Investigations Report,” Verizon, 2024. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[29] S. Majumdar et al., “ProSAS: Proactive security auditing system for clouds,” *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1219–1232, 2019.

[30] H. Takabi, J. B. D. Joshi, and G.-J. Ahn, “Security and privacy challenges in cloud computing environments,” *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010.

[31] M. D. Ryan, “Cloud computing security: The scientific challenge, and a survey of solutions,” *Journal of Systems and Software*, vol. 86, no. 9, pp. 2263–2268, 2013.

[32] N. Kshetri, “Privacy- and security-related policy issues in cloud computing,” *Telecommunications Policy*, vol. 37, no. 4–5, pp. 372–386, 2013.

[33] K. Dahbur, B. Mohammad, and A. B. Tarakji, “A survey of risks, threats and vulnerabilities in cloud computing,” in *Proc. Int. Conf. on Intelligent Semantic Web-Services and Applications (ISWSA)*, New York: ACM, 2011.

[34] N. Gonzalez, C. Miers, F. Rerêda, M. Simplicio, T. Carvalho, M. Naslund, and M. Pourzandi, “A quantitative analysis of current security concerns and solutions for cloud computing,” *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 1, no. 1, p. 11, 2012.

[35] S. Bleikertz, C. Vogel, and T. Groß, “Cloud radar: Near real-time detection of security failures in dynamic virtualized infrastructures,” in *Proc. 30th Annual Computer Security Applications Conference (ACSAC 2014)*, New York: ACM, 2014, pp. 26–35.

[36] L. M. Kaufman, “Data security in the world of cloud computing,” *IEEE Security & Privacy*, vol. 7, no. 4, pp. 61–64, 2009.

[37] M. Almorsy, J. Grundy, and A. S. Ibrahim, “Collaboration-based cloud computing security management framework,” in *Proc. IEEE 3rd Int. Conf. on Cloud Computing (CLOUD)*, IEEE, 2010, pp. 364–371.

[38] A. Behl, “Emerging security challenges in cloud computing: An insight to cloud security challenges and their mitigation,” in *Proc. World Congress on Information and Communication Technologies (WICT)*, IEEE, 2011, pp. 217–222.

[39] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos, “Security and privacy for storage and computation in cloud computing,” *Information Sciences*, vol. 258, pp. 371–386, 2014.

[40] K. Popovic and Z. Hocenski, “Cloud computing security issues and challenges,” in *Proc. 33rd Int. Convention MIPRO*, IEEE, 2010, pp. 344–349.

[41] E. B. Fernandez, “Security in the cloud: A new approach using security patterns and law compliance,” in *Proc. 9th Int. Conf. on Availability, Reliability and Security (ARES 2014)*, IEEE, 2014, pp. 534–539.

[42] D. Ardagna, A. Capobianco, M. Ciavotta, E. Gianniti, and C. Guerriero, “Multi-cloud continuous compliance,” in *Proc. IEEE Int. Conf. on Cloud Engineering (IC2E)*, IEEE, 2015, pp. 203–208.

[43] S. K. Garg, S. Versteeg, and R. Buyya, “A framework for ranking of cloud computing services,” *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.

[44] T. Garfinkel and M. Rosenblum, “When virtual is harder than real: Security challenges in virtual machine based computing environments,” in *Proc. 10th Workshop on Hot Topics in Operating Systems (HotOS X)*, USENIX, 2005.

[45] C. Onwubiko, “Security issues to cloud computing,” in N. Antonopoulos and L. Gillam (Eds.), *Cloud Computing: Principles, Systems and Applications*. Springer, 2010, pp. 271–288.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[46] D. Puthal, B. P. S. Sahoo, S. Mishra, and S. Swain, "Cloud computing features, issues, and challenges: A big picture," in *Proc. Int. Conf. on Computational Intelligence and Networks (CINE)*, IEEE, 2015, pp. 116–123.

[47] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[48] S. Ali et al., "Security and privacy in multi-cloud and hybrid cloud environments: Challenges, strategies, and future directions," *Computers & Security*, vol. 157, p. 104599, 2025. DOI: 10.1016/j.cose.2025.104599.

[49] J. W. Rittinghouse and J. F. Ransome, *Cloud Computing: Implementation, Management, and Security*. CRC Press, 2016.

[50] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: Towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2008.

[51] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST Special Publication 800-145, National Institute of Standards and Technology, Gaithersburg, MD, 2011.

[52] D. Baur et al., "Cloud resource orchestration in the multi-cloud landscape: A systematic review of existing frameworks," *Journal of Cloud Computing*, vol. 8, p. 23, 2019. DOI: 10.1186/s13677-019-0138-7.

[53] R. Gupta et al., "Cloud security posture management: Automating risk detection, compliance enforcement, and vulnerability remediation," in *Proc. IEEE Int. Conf. on Cloud Engineering (IC2E 2024)*, IEEE, 2024.

[54] A. Piplai, S. Mittal, A. Joshi, T. Finin, J. Holt, and R. Zak, "Creating cybersecurity knowledge graphs from malware after action reports," *IEEE Access*, vol. 8, pp. 211691–211703, 2020.

[55] T. Pahi, M. Leitner, and F. Skopik, "Analysis and assessment of situational awareness models for national cyber security centers," in *Proc. ICISSP 2017*, SciTePress, 2017, pp. 334–345.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Isogeny-Based Identification Using Modified Schnorr Sigma Protocol

Mahdi Mahdavi[1], Zaira Pindado[2], Helena Rifà-Pous[1], and Amineh Sakhaie[3]
[1] Universitat Oberta de Catalunya, UOC-Tech, [2] Barcelona Supercomputing Center (BSC), [3] University of Lisbon
e-mail: {m_mahdavi | hrifa}@uoc.edu zaira.pindado@bsc.es
fc65105@alunos.ciencias.ulisboa.pt

***Abstract*—Quantum computing threatens classical cryptographic protocols like the Schnorr identification scheme, which relies on the Discrete Logarithm Problem (DLP), vulnerable to quantum attacks. In this paper, we propose a modification to the classical Schnorr protocol by redefining the prover response as $r = cu \pm x \bmod q$ instead of $r = u + cx \bmod q$. While this adjustment preserves the arithmetic simplicity of the original protocol, it introduces subtle but significant changes to the protocol's security and verifiability. We analyze its soundness, zero-knowledge properties, extractor functionality, and practical viability, and explore its adaptation into a secure digital signature system under standard cryptographic assumptions. To underscore the practical significance of our approach, we implement the modified protocol within an isogeny-based framework, demonstrating its capacity to enhance an existing identification scheme with respect to both security and efficiency. Our findings illustrate that revisiting classical protocols through judicious modifications can yield more robust, quantum-resistant solutions for applications like blockchain. Finally, we present an MPC-in-the-head version of our isogeny-based protocol that allows us to rely on the standard Isogeny DLP assumption and reduce both computational and communication overhead. As far as we know it is the first isogeny-based construction with MPC-in-the-head.**

Keywords-Schnorr Protocol; Zero-knowledge proofs; Discrete Logarithm Problem; Isogeny-based cryptography; Post-Quantum Cryptography.

I. Introduction

This paper is an extended version of [1], previously published in the SECURWARE Conference. Identification protocols are fundamental cryptographic primitives that enable a prover to convince a verifier of their identity by demonstrating knowledge of a secret without disclosing it. These protocols are the foundation for many cryptographic systems, such as authentication frameworks, zero-knowledge proofs, and digital signature algorithms. One of the most celebrated and widely studied identification schemes is the Schnorr identification protocol [2], which offers a simple and elegant construction grounded in the hardness of the Discrete Logarithm Problem (DLP) [3][4]. This assumption and related computational hard problems play a central role in the design of various security protocols and schemes, such as Private Set Intersection (PSI) [5][6][7] and Attribute-Based Encryption (ABE) [8][9][10].

The Schnorr protocol is a canonical Σ-protocol that defines the prover's response to the verifier's challenge c as $r = u + cx \bmod q$, where u is a random nonce used in the commitment, x is the prover's secret and c is the challenge. This formulation balances efficiency and security, and forms the basis for many digital signature schemes through the Fiat–Shamir transformation [11].

As a Σ-protocol, Schnorr satisfies three essential properties: completeness, which ensures that an honest prover always convinces the verifier; special soundness, which guarantees that if an adversary can produce two accepting transcripts with the same commitment but different challenges, then it is possible to efficiently extract the secret x, and Honest Verifier Zero-Knowledge (HVZK), meaning that a simulator, given access to the challenge, can generate transcripts that are computationally indistinguishable from real interactions, without knowing the prover's secret.

Beyond the DL problem, the integer factorization problem is also widely used for identification proofs [12]. Factorization-based assumptions are further utilized in developing trustworthy cryptographic algorithms [13] as well as Homomorphic Encryption schemes [14][15].

A. Related Works

The Schnorr protocol has been extensively studied and extended in various directions. Fuchsbauer et al. [16] analyzed blind Schnorr signatures and signed ElGamal encryption techniques using the Algebraic Group Model (AGM), demonstrating robust security guarantees under normal assumptions without the use of heuristic arguments.

In the threshold setting, Bacho et al. [17] introduced HARTS, the first threshold Schnorr signature scheme that is simultaneously adaptively secure, robust under full asynchrony, and communication-efficient. HARTS supports high-threshold configurations—where the number of required signers can significantly exceed the corruption threshold—and outputs standard Schnorr signatures using only one asynchronous online round and subcubic communication.

Fukumitsu and Hasegawa [18] demonstrated that Schnorr signatures are secure in the multi-user setting under the AGM, assuming the hardness of the DLP. This multi-user resilience is essential for large-scale deployments, such as public key infrastructures.

In parallel, Fuchsbauer and Wolf [19] proposed a practical, concurrently secure blind signature protocol compatible with standard Schnorr signatures. Their technique ensures system compatibility while introducing predicate blind signatures, enabling signers to impose constraints on signed messages—a feature particularly valuable for privacy-preserving blockchain applications [20].

In post-quantum cryptography, Galbraith, Petit, and Silva [21] developed two digital signature systems based on the hardness of isogeny problems over supersingular elliptic curves, leveraging a novel identification technique to achieve

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

quantum-resistant security. A key innovation in their work is a novel identification technique that builds upon a well-established computational problem but addresses limitations seen in prior methods. These systems can be converted into secure digital signatures using both classical and quantum-safe approaches, providing a realistic path to efficient post-quantum cryptography solutions. In a related advancement, Baghery et al. [22] adapted the Schnorr sigma protocol to the isogeny-based setting.

These developments illustrate the robustness of the Schnorr paradigm across diverse cryptographic settings. Building on this foundation, we introduce a novel algebraic modification to the protocol's response function to improve efficiency and resilience in isogeny-based, post-quantum identification schemes. In our latest construction, we tackled the challenge of developing an Isogeny-based ID protocol based on the Isogeny Discrete Logarithm problem, which involved a larger challenge space, by employing an MPC-in-the-head construction within Isogeny-based protocols.

B. Our Contribution

In this paper, we propose a novel variation of the Schnorr identification protocol in which the prover's response is computed as $r = cu \pm x \mod q$, fundamentally altering the interaction between the nonce, challenge, and secret. This structural change results in a new verification equation and requires a complete re-evaluation of the protocol's security properties. Unlike minor tweaks, our modification challenges the conventional structure and allows for new analytical insights.

Our main contributions can be summarised as follows.

1) Formal definition and analysis of the modified protocol, reversing the typical dependency between the challenge and the secret. We also prove the security proofs that guarantee the protocol maintains completeness, special soundness, and honest-verifier zero-knowledge.
2) We analyze how the modified response can be adapted for use in non-interactive settings via the Fiat–Shamir heuristic, preserving signature viability.
3) We propose a new post-quantum id protocol based on isogenies using the modified Schnorr protocol.
4) An examination of the proposed Sigma protocol's application within isogeny-based cryptographic systems. Building upon and extending prior work [22], we identify significant advantages, most notably the elimination of the requirement for witnesses at critical proof stages. This refinement enhances protocol resilience by preventing leakage of errors related to the protocol, or witness during execution.
5) An MPC-in-the-head version of the isogeny-based Sigma protocol that reduces its computational and communication overhead and relies on the standard Isogeny DLOG assumption.

In addition, our modified Schnorr protocol enables the use of the MPC-in-the-Head technique and its advantages, which we leave as an avenue for future work.

This paper is structured as follows. In Section II, we provide the necessary background on Σ-protocols, digital signature schemes, and isogeny-based identification systems. Section III introduces our modified Schnorr protocol in detail, including the new response format and its implications on completeness, special soundness, and honest-verifier zero-knowledge. We also show how our construction leads to a secure digital signature scheme under the Discrete Logarithm Problem and supports non-interactive instantiations via the Fiat–Shamir transform. In Section IV, we extend the modified protocol to an isogeny-based setting, presenting two novel identification schemes that improve upon previous work by eliminating the need for witnesses during critical stages and enhancing resilience against execution errors. The first relies on the Vectorization Problem and the second is an MPC-in-the-head version that relies on the standard Isogeny DLOG assumption. The last construction leverages the Multiplication protocol [23][24] as a blackbox. Finally, Section V concludes with a summary of our findings and outlines potential directions for future research in post-quantum cryptography.

II. Preliminaries

A. Notation

Let $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ denote the set of integers modulo q, being q a positive prime integer and $\mathbb{Z}_q^*$ its multiplicative set. Let $\mathbb{G}$ be a group of order q with generator $g \in \mathbb{G}$.

Let $\mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z}$ denote the ring of integers modulo N, where N is a composite integer with a known prime factorization $N = \prod_{i=1}^{m} q_i^{r_i}$, such that $q_1 < q_2 < \cdots < q_m$ are distinct prime numbers and each $r_i \in \mathbb{N}$.

For any set S, the notation $a \xleftarrow{\$} S$ indicates that the element a is sampled uniformly at random from S. A function $\mu(X)$ from the natural numbers to the non-negative real numbers is *negligible* if for every positive polynomial p there is a constant C such that for all integers $x > C$, we have $\mu(x) < \frac{1}{p(x)}$ [22]. We denote by λ the security parameter.

Discrete Logarithm Problem (DLP)[4]. *Given a group $\mathbb{G}$, a generator $g \in \mathbb{G}$ and some element $h = g^x \in \mathbb{G}$, recovering x is called the Discrete Logarithm Problem.*

B. Sigma Protocols

Let $V = V(\lambda)$ and $W = W(\lambda)$ be two sets defined with respect to a security parameter λ. Let $\mathrm{R} \subseteq V \times W$ be a **relation** on $V \times W$ that defines a **language** $\mathrm{L} = \{v \in V : \exists w \in W, \mathrm{R}(v; w) = 1\}$. An element $w \in W$ such that $\mathrm{R}(v; w) = 1$ for some $v \in \mathrm{L}$ is called a **witness** for v.

A sigma-protocol (Σ-protocol) for the relation R is a three-round interactive protocol between two Probabilistic Polynomial-Time (PPT) algorithms: a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. The prover holds a witness w for $v \in \mathbf{L}$, and the verifier knows v. The protocol proceeds as follows: $\mathcal{P}$ sends a *commitment* a, $\mathcal{V}$ answers with a *challenge* c, and $\mathcal{P}$ sends a *response* r. The verifier accepts or rejects the proof based on the triple (a, c, r), which is called a transcript of the Σ-protocol.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

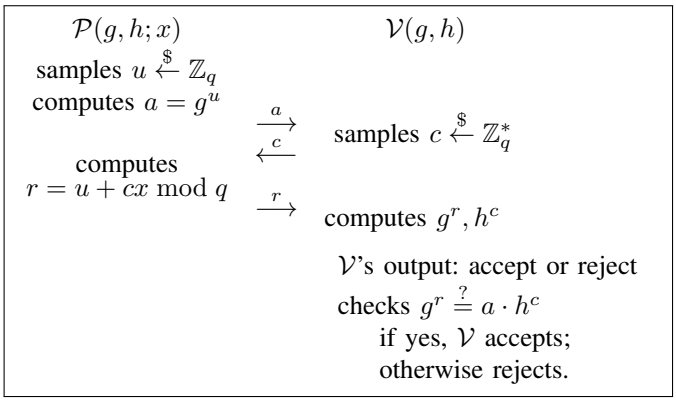


Figure 1. Protocol 1- Schnorr Protocol for relation $\mathrm{R_{ID}}$.

A Σ-protocol satisfies three properties: completeness, special soundness, and honest verifier zero-knowledge (HVZK).

Completeness. *A Σ-protocol Π with parties $(\mathcal{P}, \mathcal{V})$ is complete for* R*, if for all $(v; w) \in \mathrm{R}$, the honest $\mathcal{V}$ always accepts the honest proof of $\mathcal{P}$.*

Special Soundness. *A Σ-protocol Π has a* special soundness *for* R *if there exists a PPT extractor $\mathcal{E}$ such that, for any $v \in \mathrm{L}$, given two valid transcripts (a, c, r) and (a, c', r') with the same commitment a but different challenges $c \neq c'$, the extractor $\mathcal{E}(a, c, r, c', r')$ outputs a valid witness w such that $(v; w) \in \mathrm{R}$.*

Honest-Verifier Zero-Knowledge (HVZK). *A Σ-protocol Π satisfies* HVZK *for* R *if there exists a PPT simulator* Sim *such that, for all $(v; w) \in \mathrm{R}$, the transcript (a, c, r) generated by $\mathsf{Sim}(v)$ is computationally indistinguishable from a real transcript produced by an honest execution between the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ on input $(v; w)$.*

1) Schnorr Identification Protocol: The Schnorr protocol instantiates a Σ-protocol for the relation $\mathrm{R_{ID}} = \{(g, h; x) \mid h = g^x\}$ with $\mathbb{G}$ a multiplicative group of order q and generator g. Let $x \in \mathbb{Z}_q$ be the secret key (witness) and $h = g^x \in \mathbb{G}$ be the public key. Protocol 1 (see Figure 1) describes the steps of the Schnorr protocol.

The Schnorr protocol satisfies the key properties of a Σ-protocol:

- **Completeness:** If the prover is honest and knows x, then
$$g^r = g^{u+cx} = g^u \cdot (g^x)^c = a \cdot h^c$$
and the verifier accepts.
- **Special Soundness:** Given two accepting transcripts (a, c, r) and (a, c', r') with $c \neq c'$, we show how the verifier can extract x.First, we have $g^r = a \cdot h^c$ and $g^{r'} = a \cdot h^{c'}$. Then, by operating these two expressions, $g^r g^{-r'} = h^{c-c'}$ we get that the discrete logarithm of h is equal to $(r - r')(c - c')^{-1} \mod q$.
- **HVZK:** There exists a simulator S that, chooses random values $c \xleftarrow{\$} \mathbb{Z}_q^*$ and $r \xleftarrow{\$} \mathbb{Z}_n$, computes:
$$a = g^r \cdot h^{-c}$$
and outputs a transcript (a, c, r) that is indistinguishable from a real one.

C. Digital Signature Scheme

A **digital signature scheme** Π consists of three probabilistic polynomial-time (PPT) algorithms:

- $\mathsf{KeyGen}(1^\lambda)$: On input the security parameter λ, outputs a key pair (pk, sk), where pk is the public key and sk is the secret key.
- $\mathsf{Sign}_{sk}(m)$: On input a message $m \in \{0,1\}^*$ and the secret key sk, outputs a signature σ.
- $\mathsf{Verify}_{pk}(m, \sigma)$: On input a message m and a signature σ, outputs 1 if σ is valid for m under pk, and 0 otherwise.

We require that the scheme satisfies **correctness**, i.e., for all $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and all $m \in \{0,1\}^*$, we have:
$$\mathsf{Verify}_{pk}(m, \mathsf{Sign}_{sk}(m)) = 1.$$

Definition. *A digital signature scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is said to be **existentially unforgeable under a chosen-message attack (EUF-CMA)** if for every probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the following advantage is negligible:*

$$\mathsf{Adv}^{\mathsf{uf\text{-}cma}}_{\Pi,\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l}(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda);\\ (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\cdot)}(pk) :\\ \mathsf{Verify}_{pk}(m^*, \sigma^*) = 1\\ \wedge\ m^* \notin \mathcal{Q}\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Here:

- $\mathcal{A}^{\mathsf{Sign}(\cdot)}$ is given access to the signing oracle;
- $\mathcal{Q}$ is the set of messages queried by $\mathcal{A}$ to the signing oracle;
- $\mathsf{negl}(\lambda)$ is a negligible function in the security parameter λ.

The adversary's goal is to output a valid signature on a *new* message $m^* \notin \mathcal{Q}$ that it has not seen signed before.

D. Isogeny-Based ID Protocol Using Structured Public Keys

An identification (ID) protocol allows a prover to demonstrate the knowledge of a secret key corresponding to a public key, often formalized as a Σ-protocol over a hard relation. In isogeny-based cryptography, the underlying hardness assumption is the difficulty of computing isogenies between supersingular elliptic curves, a post-quantum hard problem.

Commutative Supersingular Isogeny Diffie–Hellman (CSIDH) [25] was proposed to enhance the efficiency of isogeny-based cryptography by using supersingular elliptic curves over the prime field $\mathbb{F}_p$. The CSI-FiSh signature scheme [26] was developed within the CSIDH framework to provide efficient isogeny-based signatures. It began with a

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

binary challenge space and was later optimized with larger public keys and an improved identification protocol, achieving subsecond signing times.

In [22], the authors propose an efficient isogeny-based identification protocol that extends CSI-FiSh [26], which was previously enhanced in [27] and [28] to support a larger challenge space through the use of structured public keys. This enhancement significantly reduces the soundness error and communication overhead. The protocol is built on Hard Homogeneous Spaces and introduces exceptional and superexceptional sets to ensure extractability and security. A non-interactive signature version is derived via the Fiat–Shamir transform, achieving strong unforgeability in the quantum random oracle model. Additionally, they present trustless key generation techniques using zero-knowledge proofs of well-formedness, making the scheme both efficient and suitable for postquantum cryptographic applications.

Let E_0 be a fixed supersingular elliptic curve over $\mathbb{F}_p$, and let $\mathrm{Cl}(\mathcal{O})$ be the class group of its endomorphism ring $\mathcal{O}$. This group acts freely and transitively on the isogeny class of E_0, defining a Hard Homogeneous Space (HHS). The secret key is an element $x \in \mathbb{Z}_N$, and the public key is $E_1 = [x]E_0$, where $[x]$ denotes the group action via an ideal class.

Classical isogeny-based ID protocols, such as those underlying CSI-FiSh [26], suffer from efficiency issues due to binary challenge spaces. To reduce the soundness error ϵ, they must be repeated λ times, where $\epsilon = 2^{-\lambda}$. The new protocol extends the challenge space to k elements, reducing the soundness error per round to $1/k$, or $1/(2k-1)$ when symmetry (through twisting) is used.

The security of our protocol is based on a hardness assumption: the $(c_0, \ldots, c_{k-1})$-*Vectorization Problem with Auxiliary Inputs*. Detailed definitions and explanations of this issue are presented in [22], which, it should be noted, draws inspiration from papers [29] and [30]. Given a starting curve E_0 and a sequence of images $\{E_i = [c_i x]E_0\}$, where $c_0 = 0$, $c_1 = 1$ and all pairwise differences $c_i - c_j$ (for $i \neq j$) are invertible modulo N. the problem is to recover the secret scalar x, under the assumption that each $c_i \in \mathbb{Z}_N$ and all pairwise differences $c_i - c_j$ are invertible modulo N. This assumption generalizes the discrete logarithm problem with auxiliary inputs to the setting of isogenies and hard homogeneous spaces.

The protocol presented in [22] can be made non-interactive using the Fiat-Shamir transform in the Quantum Random Oracle Model (QROM). This structure enables a tradeoff: larger public keys allow shorter proofs and 14× faster executions than repeated binary-challenge protocols, without compromising post-quantum security or requiring trusted third parties.

E. Hard Homogeneous Space

A Hard Homogeneous Space (HHS), as formulated by Couveignes [27], comprises a finite abelian group $\mathbb{G}$ and a finite set $\mathcal{E}$, equipped with an efficient computable group action

$$\star : \mathbb{G} \times \mathcal{E} \to \mathcal{E}.$$

This action satisfies the following structural properties:

1) **Freeness and Transitivity:** The group action is *free*, which means that for any $E \in \mathcal{E}$ and $g \in \mathbb{G}$, if $g \star E = E$, then g is the identity in $\mathbb{G}$. It is also *transitive*, to ensure that for every pair $E_1, E_2 \in \mathcal{E}$, there exists a $g \in \mathbb{G}$ such that $g \star E_1 = E_2$.
2) **Efficient Operations:** The group operation in $\mathbb{G}$, membership and equality checks in both $\mathbb{G}$ and $\mathcal{E}$, and the group action $\star$ are all efficiently computable. Furthermore, each element in $\mathbb{G}$ has a unique representation that can be computed efficiently, and elements of $\mathbb{G}$ can be sampled uniformly at random.
3) **Hard Computational Problems:** Security in HHS-based cryptosystems relies on the intractability of two core problems:
   - *Vectorization Problem:* Given $E_1, E_2 \in \mathcal{E}$, compute $g \in \mathbb{G}$ such that $g \star E_1 = E_2$.
   - *Parallelization Problem:* Given $E_1, E_2, F_1 \in \mathcal{E}$ with $E_2 = g \star E_1$ for some unknown $g \in \mathbb{G}$, compute $F_2 = g \star F_1$.

In the common special case where $\mathbb{G}$ is a cyclic group of known order N with generator g, the action can be expressed using additive notation as $[a]E := g^a \star E$, where $a \in \mathbb{Z}_N$ and $E \in \mathcal{E}$. This representation satisfies the compositional property

$$[a][b]E = [a + b]E,$$

which is frequently exploited in isogeny-based protocols.

Isogeny Discrete Logarithm Problem (IDLP). *Given a starting curve E_0 and an image $\{E_x = [x]E_0\}$, recovering the secret scalar x is called the Isogeny Discrete Logarithm Problem (IDLP).*

III. Modified Schnorr Protocol

In this section, regarded as the most significant portion of this work, we first present the new protocol based on Schnorr where the response form is specifically one of $r = cu + x$ or $r = cu - x$, previously agreed on between the prover and the verifier. We then proceed to investigate and prove its core properties.

A. Modified Schnorr Protocol

Let $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order q and g a generator. The following sigma protocol lets the prover convince a verifier about the prover's knowledge of their secret key is $x \in \mathbb{Z}_q$, such that the corresponding public key is $h = g^x \in \mathbb{G}$. More precisely, the sigma protocol is a proof system for the following relation:

$$\mathrm{R}_{\mathrm{ID}} = \{(g, h; x) : h = g^x\}.$$

The protocol for relation R_{ID} is described in Protocol 2 (see Figure 2) in the following, where previously $r = cu + x$ or $r = cu - x$ has been chosen, but use both in the exposition:

The commitment and the challenge steps of our protocol are exactly as Schnorr's. We changed the response of the prover that now is $r = cu \pm x \mod q$, which is send r to V in the final step. In the verification phase, the verifier accepts if $g^r = a^c \cdot h$ in the case of $r = cu + x$, or if $g^r \cdot h = a^c$ in the case of $r = cu - x$.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

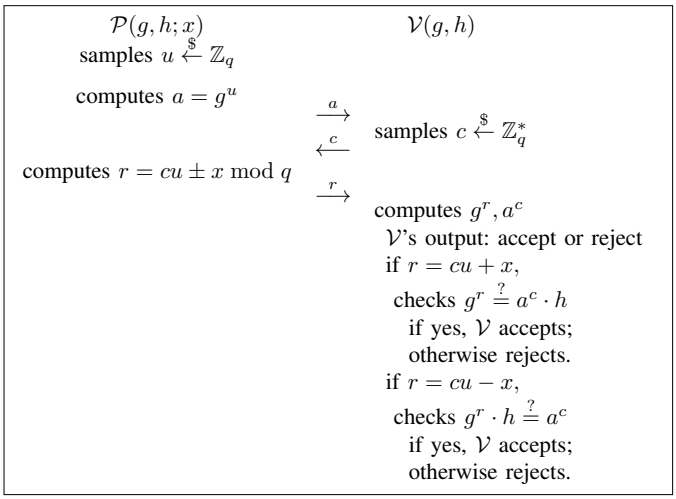


Figure 2. Protocol 2- Modified Schnorr Protocol for relation $\mathrm{R_{ID}}$.

In the following, we analyse that our modified protocol holds same properties as the original scheme that are the main properties of a sigma protocol.

1) Completeness: Let us ensure that an honest prover always passes the verifier's check.

$$g^r = g^{cu+x} = g^{cu} \cdot (g^x) = (g^u)^c \cdot (g^x) = a^c \cdot h.$$

Hence $g^r = a^c \cdot h$, So, the verifier will accept. Thus, completeness holds. For the case $r = cu - x$, the completeness property is established as follows.

$$g^r \cdot h = g^{cu-x} \cdot (g^x) = g^{cu} = (g^u)^c = a^c.$$

2) Special Soundness: To prove *soundness*, we must demonstrate that if an adversary can produce valid responses to two distinct challenges $c \neq c'$ for the same commitment a, then the secret x can be extracted.

Assume the adversary outputs a valid transcript (a, c, r). By rewinding the adversary with the same commitment a but a different challenge c', we obtain a second valid transcript (a, c', r'). Therefore, we have two valid transcripts: (a, c, r), (a, c', r') with $c \neq c'$, and both satisfying the verification equations:

$$g^r = a^c \cdot h \quad \text{and} \quad g^{r'} = a^{c'} \cdot h.$$

We compute

$$(g^r)^{c'} (g^{r'})^{-c} = (a^c h)^{c'} (a^{c'} h)^{-c} = h^{c'-c}. \tag{1}$$

Thus, the secret, which is the Discrete Logarithm of h in the basis g, can be extracted by computing:

$$x = (rc' - r'c) \cdot (c' - c)^{-1} \mod q.$$

This confirms that knowledge of valid responses to two distinct challenges allows extraction of the secret witness x, thereby proving the soundness of the protocol.

Note that we used the verification of the case that the response is, $r = cu + x$, to prove the above equality. In the case where the response is $r = cu - x$, the same relation can be employed to extract x.

3) Honest-Verifier Zero-Knowledge (HVZK): To prove HVZK, we show that the simulator chooses a random challenge c and a random response r, can simulate a valid transcript without knowing the secret x. Given the public parameters g, h, the simulator begins by selecting a random challenge $c \in \mathbb{Z}_q^*$ and a random response $r \in \mathbb{Z}_q$. Then, it computes the commitment $a \in \mathbb{G}$ so that the final transcript (a, c, r) satisfies the verification equation of the verifier.

Specifically, the simulator sets $a = (g^r h^{-1})^{c^{-1}}$ if $r = cu + x$, or $a = (g^r h)^{c^{-1}}$ if $r = cu - x$. This ensures that the verification equation $g^r = a^c \cdot h$ is valid, even if the simulator does not know x. Since both c and r are chosen independently and uniformly at random, and the commitment a is derived deterministically, the simulator output is computationally indistinguishable from that of an honest execution. Consequently, the protocol maintains the HVZK property, assuming the hardness of the discrete logarithm problem and that c is invertible modulo q.

B. Non-Interactive Schnorr and Its Modified Variant

The Fiat–Shamir transformation [11] allows converting interactive identification protocols, such as Schnorr's [31], into non-interactive zero-knowledge proofs (NIZKs). This transformation replaces the verifier's random challenge with a deterministic output derived from a cryptographic hash function, typically modeled as a random oracle. It enables the prover to independently compute the proof without interaction, making it suitable for applications such as digital signatures and proof of key possession.

In both the classical and modified Schnorr protocols, the prover first computes a commitment $a = g^u$, where $u \in \mathbb{Z}_q$ is randomly chosen. The challenge is then generated as $c = \mathcal{H}(a, m)$, where m represents the public data (e.g., a message or context), and $\mathcal{H}$ is a cryptographic hash function. The prover computes the response r using either the classical form $r = u + cx \mod q$ or the modified form $r = cu \pm x \mod q$, depending on the protocol variant.

The final proof consists of the pair (a, r). The verifier reconstructs c from the hash and checks the validity of the proof by verifying the corresponding group equation. This non-interactive approach preserves zero-knowledge and soundness under the random oracle model.

C. Modified Schnorr Signature Scheme

The Schnorr signature scheme is a non-interactive digital signature protocol derived from the interactive Schnorr identification protocol using the Fiat–Shamir transformation. It allows a prover to sign a message by demonstrating knowledge of a secret exponent x, that is the secret key, such that $h = g^x$, without revealing x itself. The transformation replaces the interactive challenge with a cryptographic hash, turning the protocol into a practical, efficient, and widely-used digital signature scheme.

a) Setup:: Let $\mathbb{G}$ be a cyclic group of prime order q, generated by g. The prover's private key is $x \in \mathbb{Z}_q$, and the

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

corresponding public key is $h = g^x$. A cryptographic hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q$ is modeled as a random oracle.

b) Signing:: To sign a message $m \in \{0,1\}^*$:

1) Sample $u \xleftarrow{\$} \mathbb{Z}_q$, and compute $a = g^u$.
2) Compute the challenge $c = \mathcal{H}(a, m)$.
3) Compute the response $r = cu \pm x \mod q$.
4) Output the signature $\sigma = (a, r)$.

c) Verification:: To verify a signature $\sigma = (a, r)$ on a message m, compute $c = \mathcal{H}(a, m)$ and check:

$$g^r \stackrel{?}{=} a^c \cdot h, \text{ if } r = cu + x \mod q, \text{ or}$$
$$g^r \cdot a^c \stackrel{?}{=} h, \text{ if } r = cu - x \mod q.$$

If the equality holds, the signature is valid.

D. EUF-CMA Security of Modified Schnorr Signature in the Random Oracle Model

We present a straight-line simulation-based proof of existential unforgeability under chosen message attacks (EUF-CMA) for the Schnorr signature scheme in the programmable Random Oracle Model (pROM). The proof leverages the ability to program the random oracle and simulate signatures via algebraic relationships. To establish security, we rely on the Forking Lemma, a probabilistic technique introduced by Pointcheval and Stern [32]. This lemma plays a central role in proving the existential unforgeability under chosen message attacks (EUF-CMA) for digital signature schemes in the random oracle model, particularly for those derived from identification protocols like Schnorr.

Theorem 1: Assume the hardness of the Discrete Logarithm (DL) problem in a cyclic group $\mathbb{G}$ of prime order q generated by g. Then, the Modified Schnorr signature scheme is existentially unforgeable under chosen message attacks (EUF-CMA) in the programmable Random Oracle Model.

Proof. Let $\mathcal{A}$ be an efficient adversary that can produce a valid forgery for the Schnorr signature scheme with non-negligible probability ε. We construct a reduction $\mathcal{R}$ that uses $\mathcal{A}$ to solve the discrete logarithm problem. $\mathcal{R}$ receives a challenge $h = g^x$.

for unknown x, and its goal is to answer the challenger with x. $\mathcal{R}$ will simulate the Schnorr signature environment for $\mathcal{A}$. We define in the following the Random Oracle Simulator and the Signature Oracle Simulator, both run by the reduction $\mathcal{R}$.

- **Random Oracle Simulator ($\mathcal{H}(\cdot)$):** $\mathcal{R}$ maintains a table to simulate the random oracle $\mathcal{H}$. When $\mathcal{A}$ queries $\mathcal{H}(a, m)$:
  1) If (a, m) was previously queried, return the same value c.
  2) Otherwise, choose $c \in \mathbb{Z}_q$ uniformly at random, store (a, m, c), and return c.

  Later, $\mathcal{R}$ can program $\mathcal{H}(a, m)$ to any desired value c not yet used.

- **Signature Oracle Simulator ($\mathrm{Sign}(\cdot)$):** When $\mathcal{A}$ queries the signing oracle on message m:
  1) Choose random $c \in \mathbb{Z}_q^*$, $r \in \mathbb{Z}_q$.
  2) Compute a as follows.
     - if $r = cu + x$ computes $a = \left(g^r h^{-1}\right)^{1/c} \mod q$.
     - if $r = cu - x$ computes $a = \left(g^r h\right)^{1/c} \mod q$.
  3) Program the random oracle so that $\mathcal{H}(a, m) := c$.
  4) Return $\sigma = (a, r)$ as the signature.

  The signature is valid because it passes the verification equation and is indistinguishable from a genuine signature because it has the same distribution.

- **Forgery and Extraction:** Suppose $\mathcal{A}$ outputs a forgery (m^*, a^*, r^*) such that:

  $$\mathcal{H}(a^*, m^*) = c^*, \quad \text{and} \quad g^{r^*} = a^{*^{c^*}} h$$

  and (a^*, m^*) was never queried before to the signing oracle, so it programs $\mathcal{H}(a^*, m^*) = c^*$ to a value it chooses.
  The reduction $\mathcal{R}$ rewinds forger $\mathcal{A}$ and reprograms the random oracle $\mathcal{H}$ to return different values $c' \neq c^*$ when queried on (a^*, m^*). this implies obtaining another forgery (m^*, a^*, r') leading to:

  $$g^{r^*} = a^{*^{c^*}} \cdot h, \quad g^{r'} = a^{*^{c'}} \cdot h.$$

  Defining $x := \mathrm{DL}_g(h)$ and $u := \mathrm{DL}_g(a^*)$, we have that $r^* = c^* u + x$ and $r' = uc' + x$, thus similar to equation (1) $x = \frac{c' \cdot r^* - c^* \cdot r'}{c' - c^*} \mod q$.
  Thus, x (the discrete logarithm of h) is extracted, breaking the DL assumption. In the case where $r^* = c^* u - x$ and $r' = c' u - x$, the same relations can be employed to extract x.

□

The ability to forge a Schnorr signature implies solving the discrete logarithm problem, contradicting its assumed hardness. Therefore, under the programmable random oracle model and the DL assumption, the Schnorr signature is EUF-CMA secure. The reduction runs in expected polynomial time, and the probability that $\mathcal{R}$ sends to the challenger x is at least ε.

Remark. In both the original and modified Schnorr protocols, the commitment and challenge phases can be precomputed offline, independent of the secret witness, benefiting time-sensitive applications like blockchain systems. The offline phase involves computations without secret inputs, allowing the prover and verifier to execute and store protocol steps (except the response) in advance. The online phase, where the secret witness x is used, completes the protocol. In the original protocol, the response $r = u + cx$ is computed entirely online due to its dependence on x. Our modified protocol allows partial precomputation of the response ($r = cu$) offline, using the non-secret challenge c, requiring only the addition of x online, thus reducing computation and communication costs in the interactive setting. This advantage aligns with MPC

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

interpretations but diminishes in the non-interactive setting (e.g., via Fiat–Shamir transformation), where both protocols generate the full proof in a single step.

IV. Isogeny-Based ID Protocol Using Modified Schnorr

The isogeny-based identification protocol presented in [22] extends the CSI-FiSh framework by introducing structured public keys, which significantly improve efficiency and reduce the soundness error rate. Although it operates within the Hard Homogeneous Space (HHS) formed by the class group acting on supersingular elliptic curves, the protocol maintains a classic Σ-protocol format with a commitment, challenge, and response reminiscent of the Schnorr identification scheme.

Our modified Isogeny-Based ID Schnorr variant offers a conceptual change by redefining the prover response as $r = cu \pm x \mod q$, in contrast to the traditional $r = u + cx \mod q$. The resulting protocol has completeness, special soundness, and HVZK, making it suitable for efficient Fiat-Shamir-based signature schemes.

A. An Efficient ID Protocol Based on Modified Schnorr

Let p be a large prime such that the supersingular elliptic curves over $\mathbb{F}_p$ form a well-connected isogeny graph. Denote by $\mathcal{E}$ the set of $\mathbb{F}_p$-isomorphism classes of supersingular elliptic curves, and let $\mathrm{Cl}(\mathcal{O}) \cong \mathbb{Z}_N$ denote the class group of maximal order $\mathcal{O}$ in a quaternion algebra acting on $\mathcal{E}$ via isogenies.

The pair $(\mathbb{Z}_N, \mathcal{E})$ thus defines a *hard homogeneous space* [27] $(\mathcal{G}, \mathcal{X})$, equipped with a free and transitive action:

$$[a] \star E := \phi_a(E), \quad \text{for } a \in \mathbb{Z}_N,\ E \in \mathcal{E}.$$

Let $E_0 \in \mathcal{E}$ denote a publicly agreed base curve. We assume the existence of a publicly known *exceptional set* $C = \{c_0 = 0, c_1 = 1, \ldots, c_{k-1}\} \subset \mathbb{Z}_N$ such that every pairwise difference $c_i - c_j \in \mathbb{Z}_N^*$ is invertible. This assumption enables the construction of a sound Σ-protocol with extractability. Based on [22], we know that an Exceptional Set is defined as follows.

Definition. *Let $N \in \mathbb{Z}_{>0}$. A subset $C = \{c_0, c_1, \ldots, c_{k-1}\} \subset \mathbb{Z}_N$ is called an* exceptional set modulo N *if all pairwise differences are invertible, i.e., for all $i \neq j$, the element $c_i - c_j \in \mathbb{Z}_N^*$. This set guarantees that for any two distinct challenges $c, c' \in \mathbb{C}$, the value $c - c'$ is invertible modulo N, enabling efficient extraction in Σ-protocols.*

Remark. Given a target size k for an exceptional set and a modulus N, it is sufficient that the smallest prime factor q_1 of N satisfies $q_1 \geq k$. Under this condition, there exists an efficient algorithm, referred to as `XSGen`, capable of generating an exceptional set $C = \{c_0, c_1, \ldots, c_{k-1}\} \subseteq \mathbb{Z}_N$ of size k, in which all pairwise differences $c_i - c_j$ (for $i \neq j$) are invertible modulo N. If $q_1 < k$, one can still construct such a set by restricting the operation to a subgroup of $\mathbb{Z}_N$ in which smaller prime divisors are eliminated. This involves factoring out those small primes so that the minimal prime factor of the resulting subgroup is at least k. The only structural constraint imposed on N is that it must not be k-smooth, that is, N should not be composed entirely of prime factors less than k, which is typically a reasonable assumption for cryptographic applications involving large composite moduli [22].

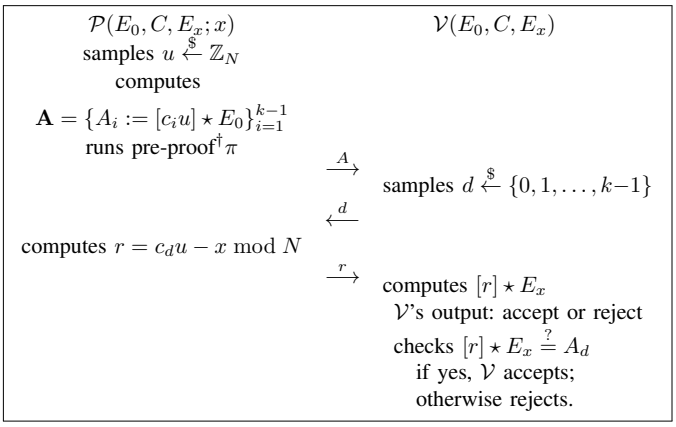


Figure 3. Protocol 3- Isogeny Schnorr Protocol for relation $\mathrm{R_{ID}}$.

B. Identification Protocol Steps

In the first step for this protocol, we start with Key Generation. The prover samples $x \xleftarrow{\$} \mathbb{Z}_N$ as a secret key, and the public key is $E_x := [x] \star E_0$. The tuple $(E_0, E_x) \in \mathcal{E}^2$ is published, while x remains private to the prover. The identification protocol is a 3-move public-coin Σ-protocol defined by the protocol 3 (see Figure 3) as following steps. This Protocol ensures that the prover demonstrates knowledge of x consistent with the challenge and commitment.

Before detailing the protocol's primary characteristics, it is essential to first outline its underlying structure and associated benefits. Building on the proof techniques presented in [22], it can be readily shown that the set $\mathbf{A}$ in protocol 3 is well-defined and retains its essential properties. For any A_i, a corresponding proof, referred to as a pre-proof, must be provided. This pre-proof follows the same approach as the proof techniques in [22], concretely Theorems 5.1 and 5.2 in [22], as well as remark (IV-A).

It is important to note that the protocol presented in [22] operates under idealized assumptions—namely, that all components, including randomness and network integrity, function flawlessly. Under such conditions, any deviation—such as inaccurate randomness or communication failures— during pre-proof can compromise the witness and, consequently, the security of the main protocol by causing information leakage from x.

On the other hand, generating all $E_i = [c_i x]E_0$ in [22] requires invoking the aforementioned theorems and consistently relying on the witness x during the pre-proof construction. However, in the modified isogeny-based Schnorr protocol, this reliance is mitigated by replacing x with u, thereby reducing direct dependence on the witness. In contrast, our modified Schnorr protocol exhibits greater resilience, enabling corrective measures to be taken without undermining its core functionality. More specifically, in the modified isogeny-based Schnorr protocol, it is sufficient to halt execution, select a new value u, and restart the pre-proof and protocol—without

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

affecting the witness. This property significantly enhances the protocol's reliability under failure scenarios or adversarial conditions.

1) Completeness: We know that if the prover follows the protocol honestly, the verifier accepts with probability 1. For our modified Schnorr protocol, given $A_d = [c_d u] \star E_0$, $E_x = [x] \star E_0$, and $r = c_d u - x$, we compute:

$$[r] \star E_x = [c_d u - x] \star [x]E_0 = [c_d u] \star E_0 = A_d.$$

Therefore, the verifier check passes.

2) Special Soundness: We demonstrate that given two valid transcripts for the same commitment and distinct challenges, the prover's secret x can be recovered efficiently. Given two accepting transcripts (A, d, r) and (A, d', r') with $d \neq d'$ and a known set $C = \{c_0 = 0, c_1 = 1, c_2, \cdots, c_{k-1}\}$, we have:

$$[r]\star E_x = A_d = [c_d u]\star E_0 \quad \text{and} \quad [r']\star E_x = A_{d'} = [c_{d'} u]\star E_0$$

note that we can extract c_d and $c_{d'}$ by having d, d' and set C. From the verification equation, one can conclude that $[r]E_x = A_d$ and $[r']E_x = A_{d'}$, and from the pre-proof (or trusted) commitment we know that $A_i = [c_i u]E_0$ for $i = 1, \cdots, k-1$. These imply that we have $[r][x]E_0 = [c_d u]E_0$ and $[r'][x]E_0 = [c_{d'} u]E_0$, so:

$$[r + x]E_0 = [c_d u]E_0 \quad \text{and} \quad [r' + x]E_0 = [c_{d'} u]E_0$$

These imply that:

$$[c_{d'}(r+x)]E_0 = [c_{d'} c_d u]E_0 \quad \text{and} \quad [c_d(r'+x)]E_0 = [c_d c_{d'} u]E_0$$

Since the right part of relations are the same we have:

$$[c_{d'}(r + x)]E_0 = [c_d(r' + x)]E_0$$

It implies that

$$c_{d'} r + c_{d'} x = c_d r' + c_d x \;\Rightarrow$$
$$c_{d'} r - c_d r' = c_d x - c_{d'} x = (c_d - c_{d'})x.$$

Since $d \neq d'$ so $c_d \neq c_{d'}$. Therefore, we can divide both sides to $c_d - c_{d'} \in \mathbb{Z}_N^*$ and then we compute:

$$x = \frac{c_{d'} r - c_d r'}{c_d - c_{d'}} \mod N.$$

This proves extractability and thus special soundness.

3) Honest-Verifier Zero-Knowledge (HVZK): To prove HVZK, we construct a simulator that produces a valid-looking transcript (A, c, r) without knowing the secret key $x \in \mathbb{Z}_N$. Consider that the simulator has a sequence of images $\{E_i = [c_i x]E_0\}$ according to $(c_0, \ldots, c_{k-1})$-Vectorization Problem with Auxiliary Inputs. The simulator selects $u \in \mathbb{Z}_N$ and samples a challenge $d \in \{0, \ldots, k-1\}$, both uniformly at random and set $E_x = [c_d x]E_0 = E_d$. Given random u and sequence $\{E_i = [c_i x]E_0\}$, the simulator calculates $A = \{A_i := [c_i u] \star E_i = [c_i u + c_i x] \star E_0\}_{i=1}^{k-1}$ and sets the response as $r = c_d u$. The resulting transcript (A, d, r) satisfies the verifier check by construction $[r] \star E_x = A_d$ and is identical to a real transcript, thus establishing the HVZK property.

Remark. In [22], the pre-proof phase involves verifying the public set $E_i = [c_i x]E_0$, directly involving the secret x. In contrast, our protocol verifies $E_i = [c_i u]E_0$, where u is a random value unrelated to the secret.

If, during the pre-proof phase in [22], the randomness used in the commitment has insufficient entropy, the verifier could potentially recover the secret x from the response. In our protocol, even if such a weakness occurs, only the random value u could be exposed, without compromising x. In that case, the prover can simply discard u and any related computations, select a fresh random value, and rerun the pre-proof securely.

C. MPC in the Head Isogeny Schnorr Protocol

In this section, we explain how modifying the response from $r = u + c \cdot x$ to $r = c \cdot u + x$ in the isogeny-based version of the protocol allows the elimination of the exceptional set C. This change reduces both computational and communication overhead. Moreover, the underlying hard problem shifts from the $(c_0, \ldots, c_{k-1})$-Vectorization Problem to the Isogeny Discrete Logarithm problem, which is more standard and well-studied. In essence, our modification enables the possibility of implementing an MPC-in-the-head technique for the Isogeny-Schnorr protocol.

To achieve this, we utilize a post-quantum secure multiplication protocol based on Oblivious Transfer (OT). Such protocols enable two parties, holding secret values u and c, to jointly compute shares a and b such that $uc = a + b$, without revealing u or c to each other. We do not propose the detailed construction of the multiplication protocol and, instead, treat it as a black box within our design. Specifically, one party receives a and the other receives b, while maintaining the confidentiality of their respective shares. This mechanism will be used between the prover and the verifier, each holding their respective secrets. They do it multiple times to obtain a vector of shares each other. Later, the prover shares all of them to the verifier except one, which is going to be used in the sigma protocol. The security relies in the fact that the prover does not know which one is going to be asked to use by verifier.

Assume that the prover wants to prove knowledge of the secret x corresponding to the public key $E_x = [x] \star E_0$, where the tuple $(E_0, E_x) \in \mathcal{E}^2$ is public, and x remains private.

The protocol proceeds as follows, a first commitment phase where each party selects the secret values for the multiplication protocol. The prover sends the commitments of their selected random integers to the verifier. They then run the multiplication protocol. This protocol generates the shares $\vec{a}, \vec{b}$ for the prover and verifier, respectively. After that, the prover sends the commitments of their shares to the verifier. Then, in the cut-and-choose phase the verifier opens all the commitments except one (j) and, if they pass the multiplication protocol checking, then they start the sigma protocol, which allows the prover to prove knowledge of secret x using the shares and secrets corresponding to j. We detail the phases in the following and give the sigma protocol in 4:

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

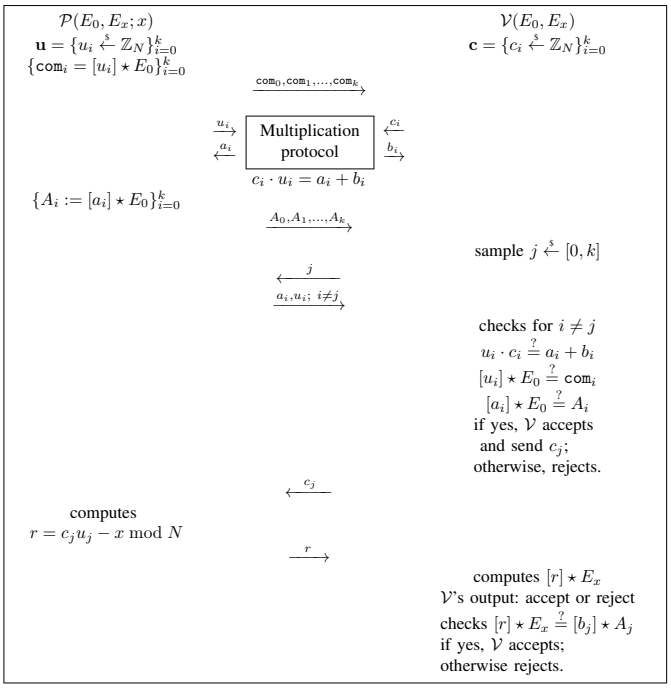


Figure 4. Protocol 4- MPC-in-the-head Isogeny Schnorr Protocol for relation $\mathcal{R}_{\text{id}}$

1) **Commitment Phase:**
   - The prover selects random values $\mathbf{u} = [u_0, u_1, \ldots, u_k]$ and publishes their commitments $\mathtt{com}_i := [u_i] \star E_0$ for each i.
   - The verifier selects random challenge values $\mathbf{c} = [c_0, c_1, \ldots, c_k] \in \mathbb{Z}_N^{k+1}$.
2) **Multiplication Protocol:**
   - For each pair (u_i, c_i), the prover and verifier engage in a multiplication protocol, obtaining shares (a_i, b_i) such that: $c_i \cdot u_i = a_i + b_i$. The prover computes and sends the commitments $A_i = [a_i] \star E_0$ for all i.
3) **Cut-and-Choose Phase:**
   - The verifier selects an index $j \in [0, k]$ and requests the prover to send all a_i and u_i except $i = j$ to open all commitments except index j. So the prover sends all a_i and u_i except $i = j$.
   - The verifier checks that for all $i \neq j$: $u_i \cdot c_i = a_i + b_i$ and verifies the corresponding commitments $[u_i] \star E_0 \stackrel{?}{=} \mathtt{com}_i$ and $[a_i] \star E_0 \stackrel{?}{=} A_i$ for all $i \neq j$.
4) **Sigma Protocol Start:**
   - If all checks pass, the verifier accepts and proceeds to run the main sigma Protocol 4 using the unopened elements: $[u_j] \star E_0$, $[a_j] \star E_0$, c_j, and b_j, knowing that: $c_j \cdot u_j = a_j + b_j$.

We can easily check that, if the prover is honest, then $[b_j]A_j = [b_j][a_j] \star E_0 = [b_j + a_j] \star E_0$ and knowing that $a_j + b_j = c_j u_j$, the last equation is equal to $[c_j u_j] \star E_0 = [r + x] \star E_0 = [r][x] \star E_0 = [r] \star E_x$. The intuition for soundness is the prover could guess the j in advance and select an appropriate number just with probability $1/(k+1)$. Then, it is safe for an enough big k. The HVZK can be simply investigated similar to the previous section.

Through this approach, the exceptional set C is no longer necessary, and the protocol relies on the hardness of the Isogeny Diffie-Hellman problem II-E. The use of MPC-in-the-head techniques significantly enhances the protocol's practicality in isogeny-based cryptography.

V. Conclusion

We have introduced a modified version of the Schnorr Sigma protocol, redefining the prover's response to reduce its dependency on the secret witness x. This seemingly minor algebraic change leads to meaningful improvements in both the structural and practical aspects of the protocol. Through a formal analysis, we have demonstrated that the modified scheme retains its fundamental security properties—including soundness and zero-knowledge—while offering enhanced robustness and flexibility.

Applying this construction in the isogeny-based setting, we addressed key limitations of an existing identification protocol, particularly those arising from its idealized assumptions and its heavy reliance on the witness during pre-proof generation. By shifting this dependency from x to a fresh random value u, our approach enables safer recovery from randomness failures or communication errors, without compromising the security of the secret key. This resilience to faults and adversarial interruptions marks a significant improvement in the protocol's practicality and reliability for real-world deployment. Finally, we proposed the first isogeny-based MPC-in-the-head identity protocol, to the best of our knowledge. This construction demonstrates that our modified Schnorr variant is particularly well-suited for adaptation to zero-knowledge protocols following the MPC-in-the-head paradigm, opening new directions for efficient post-quantum proofs based on isogenies.

Our work illustrates how carefully rethinking classical cryptographic constructions can lead to more robust solutions in post-quantum settings, such as isogeny-based cryptography, and opens the door to further exploration of protocol modifications that enhance security under realistic conditions.

Acknowledgments

This work was supported by the Spanish Ministry of Science and Innovation through the PID2021-125962OB-C31 "SECURING" project. Additional funding was provided by the ARTEMISA International Chair of Cybersecurity (C057/23) and the DANGER Strategic Project of Cybersecurity (C062/23), both funded by the Spanish National Institute of Cybersecurity through the European Union — NextGenerationEU and the Recovery, Transformation, and Resilience Plan. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. Zaira Pindado is supported by ERC grant (HomE, 101043467).

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

References

[1] M. Mahdavi, Z. Pindado, A. Sakhaie, and H. Rifà-Pous, "A modified Schnorr sigma protocol and its application to isogeny-based identification," in *SECURWARE 2025, The Nineteenth International Conference on Emerging Security Information, Systems and Technologies*, 2025, pp. 125–131, ISBN: 978-1-68558-306-4.

[2] C.-P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991. DOI: 10.1007/BF00196725.

[3] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976. DOI: 10.1109/TIT.1976.1055638.

[4] M. Mahdavi, S. Khaleghifard, and Z. Ahmadian, "New variations of discrete logarithm problem," *The ISC International Journal of Information Security*, vol. 15, no. 3, pp. 91–100, 2023, ISSN: 2008-2045. [Online]. Available: https://doi.org/10.22042/isecure.2023.184477.

[5] M. Mahdavi, M. H. Ameri, J. Mohajeri, and M. R. Aref, "A verifiable delegated set intersection without pairing," in *2017 Iranian Conference on Electrical Engineering (ICEE)*, IEEE, 2017, pp. 2047–2051. [Online]. Available: https://doi.org/10.1109/IranianCEE.2017.7985395.

[6] M. Mahdavi, M. Delavar, M. H. Ameri, J. Mohajeri, and M. R. Aref, "On the security of o-psi: A delegated private set intersection on outsourced datasets (extended version)," *The ISC International Journal of Information Security*, vol. 10, no. 2, pp. 117–127, 2018, ISSN: 2008-2045. [Online]. Available: https://www.isecure-journal.com/article_67335.html.

[7] M. Mahdavi, M. Delavar, M. H. Ameri, J. Mohajeri, and M. R. Aref, "On the security of o-psi a delegated private set intersection on outsourced datasets," in *2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, IEEE, 2017, pp. 77–81. [Online]. Available: https://doi.org/10.1109/ISCISC.2017.8488358.

[8] M. Mahdavi and Z. Ahmadian, "Fine-grained flexible access control: Ciphertext policy attribute based encryption for arithmetic circuits," *Journal of Computer Virology and Hacking Techniques*, pp. 1–14, 2022. [Online]. Available: https://doi.org/10.1007/s11416-022-00459-6.

[9] M. Mahdavi, M. H. Tadayon, M. S. Haghighi, and Z. Ahmadian, "Iot-friendly, pre-computed and outsourced attribute based encryption," *Future Generation Computer Systems*, vol. 150, pp. 115–126, 2024. [Online]. Available: https://doi.org/10.1016/j.future.2023.08.015.

[10] M. Mahdavi, Z. Ahmadian, and H. Rifà-Pous, "Classification of attribute-based encryption based on access policy contents," in *FCiR25 Acta*, ser. De Cifris Koine, vol. 8, De Cifris Press, Nov. 2025, pp. 123–136, ISBN: 979-12-81863-07-1. [Online]. Available: https://doi.org/10.69091/koine/vol-8-P14.

[11] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," *Lecture Notes in Computer Science*, vol. 263, pp. 186–194, 1987. DOI: 10.1007/3-540-47721-7_12.

[12] G. Poupard and J. Stern, "Short proofs of knowledge for factoring," in *International Workshop on Public Key Cryptography*, Springer, 2000, pp. 147–166.

[13] M. Mahdavi, N. Abapour, and Z. Ahmadian, "Trustworthy approaches to rsa: Efficient exploitation strategies based on common modulus," in *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2024, pp. 2535–2540.

[14] M. Mahdavi, E. Meamari, E. H. Beni, and M. Sheikhi, "Leveled homomorphic encryption over composite groups," in *International Conference on Cryptology in Africa*, Springer, 2025, pp. 25–50. [Online]. Available: https://doi.org/10.1007/978-3-031-97260-7_2.

[15] M. Mahdavi and H. Rifà-Pous, "One-way homomorphic encryption: A composite group approach," in *International Conference on Cryptology in Africa*, Springer, 2025, pp. 74–88. [Online]. Available: https://doi.org/10.1007/978-3-031-97260-7_4.

[16] G. Fuchsbauer, A. Plouviez, and Y. Seurin, "Blind Schnorr signatures and signed elgamal encryption in the algebraic group model," *Lecture Notes in Computer Science*, vol. 12106, pp. 63–95, 2020. DOI: 10.1007/978-3-030-45724-2_3.

[17] R. Bacho, J. Loss, G. Stern, and B. Wagner, *Harts: High-threshold, adaptively secure, and robust threshold Schnorr signatures*, To appear in: Advances in Cryptology – ASIACRYPT 2024, Lecture Notes in Computer Science, vol 15486, Springer, Singapore. Edited by KM. Chung and Y. Sasaki, 2025. [Online]. Available: https://doi.org/10.1007/978-981-96-0891-1_4.

[18] M. Fukumitsu and S. Hasegawa, "On multi-user security of Schnorr signature in algebraic group model," *Proceedings of the Tenth International Symposium on Computing and Networking Workshops (CANDARW)*, pp. 295–301, 2022. DOI: 10.1109/CANDARW57323.2022.00014.

[19] G. Fuchsbauer and M. Wolf, "Concurrently secure blind Schnorr signatures," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2024, pp. 124–160.

[20] N. Abapour, A. Goharshady, C. Dragan[1], and M. Mahdavi, "Blockchain-based economic voting with posterior security from lattices," in *Advances in Information and Computer Security: 20th International Workshop on Security, IWSEC 2025, Fukuoka, Japan, November 25–27, 2025, Proceedings*, Springer Nature, 2025, p. 247. [Online]. Available: https://doi.org/10.1007/978-981-95-4674-9_13.

[21] S. D. Galbraith, C. Petit, and J. Silva, *Identification protocols and signature schemes based on supersingular isogeny problems*, Journal of Cryptology, Volume 33, Pages 130–175, Springer, https://doi.org/10.1007/s00145-019-09316-0, 2020. [Online]. Available: https://doi.org/10.1007/s00145-019-09316-0.

[22] K. Baghery, D. Cozzo, and R. Pedersen, "An isogeny-based id protocol using structured public keys," in *IMA International Conference on Cryptography and Coding*, Springer, 2021, pp. 179–197. DOI: 10.1007/978-3-030-92641-0_9.

[23] N. Gilboa, "Two party rsa key generation," in *Annual International Cryptology Conference*, Springer, 1999, pp. 116–129.

[24] I. Haitner, N. Makriyannis, S. Ranellucci, and E. Tsfadia, "Highly efficient ot-based multiplication protocols," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2022, pp. 180–209.

[25] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes, "Csidh: An efficient post-quantum commutative group action," in *Advances in Cryptology – ASIACRYPT 2018*, ser. Lecture Notes in Computer Science, vol. 11274, Springer, 2018, pp. 395–427. DOI: 10.1007/978-3-030-03332-3_15.

[26] W. Beullens, T. Kleinjung, and F. Vercauteren, "CSI-FiSh: Efficient isogeny-based signatures through class group computations," in *Advances in Cryptology – ASIACRYPT 2019, Part I*, S. D. Galbraith and S. Moriai, Eds., ser. Lecture Notes in Computer Science, Presented at ASIACRYPT 2019, vol. 11921, Kobe, Japan: Springer, Heidelberg, Germany, Dec. 2019, pp. 227–247. DOI: 10.1007/978-3-030-34578-5_9.

[27] J.-M. Couveignes, *Hard homogeneous spaces*, Cryptology ePrint Archive, Report 2006/291, Preprint, Jul. 2006. [Online]. Available: https://eprint.iacr.org/2006/291.

[28] A. Rostovtsev and A. Stolbunov, "Public-key cryptosystem based on isogenies," *Cryptology ePrint Archive*, 2006.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[29] J. H. Cheon, "Discrete logarithm problems with auxiliary inputs," *Journal of Cryptology*, vol. 23, no. 3, pp. 457–476, Jul. 2010. DOI: 10.1007/s00145-009-9047-0.

[30] T. Kim, "Multiple discrete logarithm problems with auxiliary inputs," in *Advances in Cryptology – ASIACRYPT 2015, Part I*, T. Iwata and J. H. Cheon, Eds., ser. Lecture Notes in Computer Science, vol. 9452, Auckland, New Zealand: Springer, Nov. 2015, pp. 174–188. DOI: 10.1007/978-3-662-48797-6_8.

[31] F. Hao, *Schnorr non-interactive zero-knowledge proof*, RFC 8235, Internet Engineering Task Force (IETF), Informational, Sep. 2017. DOI: 10 . 17487 / RFC8235. [Online]. Available: https://www.rfc-editor.org/info/rfc8235.

[32] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, vol. 13, pp. 361–396, 2000.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Threat-Centric Vulnerability Management: Explainable Multi-Modal Learning for Mapping CVEs to the MITRE ATT&CK Framework

Logan McMahon
School of Electronics, Electrical Engineering and Computer Science
Queen's University Belfast, United Kingdom
e-mail: lmcmahon25@qub.ac.uk

Oluwafemi Olukoya
School of Electronics, Electrical Engineering and Computer Science
Queen's University Belfast,, United Kingdom
e-mail: o.olukoya@qub.ac.uk

***Abstract*—Automated mapping of Common Vulnerabilities and Exposures (CVEs) to the MITRE ATT&CK framework is a critical yet underexplored challenge in proactive cyber defence. Existing approaches rely primarily on CVE textual descriptions, rendering them susceptible to two fundamental limitations: incomplete or poorly structured vulnerability reports, and syntactic drift — the gradual evolution of descriptive language over time that undermines the temporal robustness of learned mappings. This work directly addresses both limitations through a systematic investigation of metadata enrichment and multimodal learning strategies for CVE-to-ATT&CK tactics mapping. By augmenting CVE descriptions with structured contextual features, including exploitability scores, software weakness classifications, attack patterns, and system identifiers, the proposed framework substantially improves mapping fidelity across both unsupervised and supervised settings. In unsupervised evaluations, metadata enrichment alone yielded a 42% to 66.7% increase in correct mappings alongside a 6% reduction in misclassifications. Under supervised conditions, the proposed SecRoBERTa model achieved a weighted F1 score of 93.47% following Optuna-based hyperparameter optimisation, representing a 14.6% improvement over the strongest baseline. The introduction of a multimodal architecture further advanced performance to 95.82% after fine-tuning, a 2.35% gain over the text-only counterpart. These findings establish that integrating structured vulnerability metadata with contextual language modelling yields mappings that are simultaneously more accurate, interpretable, and resilient to temporal drift.**

Keywords-MITRE ATT&CK; CVE; Vulnerability; Machine Learning; Data Augmentation; Threat Intelligence; XAI, Cybersecurity; Late Fusion; Multimodal Approach; Vulnerability Description; SHAP; Feature Importance; Temporal Analysis; Syntactic Drift.

I. Introduction

In our previous study [1], we introduced an approach for enriching vulnerability descriptions to improve the accuracy of mapping Common Vulnerabilities and Exposures (CVEs) to the MITRE ATT&CK framework. The present work extends this research in three key directions. First, we conduct an expanded feature analysis of the string-concatenated model, employing SHAP-inspired feature importance techniques to enhance model explainability. Second, we propose a custom multimodal SecRoBERTa architecture that integrates numeric feature vectors with the contextual representation derived from the [CLS] token of the textual input. Third, we investigate the impact of syntactic drift in vulnerability descriptions by performing a time-series analysis, thereby assessing the model's robustness to evolving and previously unseen threat patterns.

In 2025, 48,244 Common Vulnerabilities and Exposures (CVEs) were published, a 66.6% increase from 2023 and a 20.4% increase from 2024, underscoring the growing challenge organizations face in managing vulnerabilities at scale [2]. Studies show that most organizations are only able to remediate 10% to 15% of open vulnerabilities each month, leaving a persistent backlog [3]. While only an estimated 1% to 6% of CVEs are actively exploited, these few can have severe consequences [4][5]. According to the Mandiant M-Trends 2025 Report [6], vulnerability exploitation was the most common initial attack vector observed in incident response investigations, emphasizing the importance of effective vulnerability prioritization.

Since it is neither feasible nor necessary to remediate every vulnerability, organizations are shifting toward risk-based vulnerability prioritization. This approach focuses on addressing vulnerabilities that pose the greatest risk, incorporating threat intelligence to enable a threat-informed defense [7]. Central to this approach is the MITRE ATT&CK framework, a widely adopted knowledge base of adversary tactics and techniques derived from real-world observations [8][9]. Within this framework, tactics represent high-level attacker goals (such as *Initial Access* or *Persistence*), while techniques describe how those goals are achieved (e.g., *exploiting a public-facing application or executing a script*).

Mapping CVEs to MITRE ATT&CK tactics and techniques allows defenders to better understand the potential impact of unpatched vulnerabilities, prioritize them based on adversary behavior, and align vulnerability management with real threat scenarios [10]. For example, prioritizing CVEs linked to tactics like *privilege escalation* or *lateral movement* can help security teams mitigate high-impact risks more effectively. However, given the rapid growth of CVEs, manually labeling each with ATT&CK mappings is infeasible. This highlights the urgent need for automated solutions to support scalable, threat-informed vulnerability management.

Recent research [11][12][13][14][15][16][17] has increasingly focused on automating the mapping of CVEs to MITRE ATT&CK techniques. Most existing methods rely heavily on CVE descriptions, with some efforts incorporating additional data, such as Common Vulnerability Scoring System

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

(CVSS) vectors and Common Weakness Enumeration (CWE) identifiers [18][19]. However, prior studies have shown that patterns derived from CWE and CVSS can be unreliable, and vulnerability descriptions themselves are often inconsistent, incomplete, outdated, or inaccurate [18][20][21][22][23].

A major challenge remains in accurately mapping CVEs with poor-quality descriptions, as these often lack sufficient detail about exploitation methods or impact. To address these limitations, recent approaches have explored using Large Language Models (LLMs) to infer missing or unclear information, though gaps in domain knowledge constrain these methods and the complexity of vulnerability language [24].

This research proposes a comprehensive, automated approach to mapping CVEs to MITRE ATT&CK tactics, formulated as a multilabel classification problem that integrates structured data to enhance accuracy, particularly when descriptive fields are limited or ambiguous. The primary contributions of this research are as follows:

- **Extended Unsupervised Mapping Pipeline:** We adapt and expand the SMET framework [14] to operate on a larger, feature-enriched CVE dataset, using mappings from the Centre for Threat-Informed Defence [25]. By integrating pre-processed CVSS, CWE, and Common Platform Enumeration (CPE) data, the modified pipeline achieves measurable improvements in full and partial mappings without requiring labelled data.
- **Creation of an Enriched Dataset for Supervised Learning:** We compile an extended dataset by incorporating structured data from the National Vulnerability Database (NVD), Common Attack Pattern Enumeration and Classification (CAPEC), and Exploit Prediction Scoring System (EPSS). This dataset enables systematic evaluation of the contribution of each feature to the CVE-to-ATT&CK mapping process.
- **Systematic Feature Evaluation:** We perform a detailed feature importance analysis to assess the impact of each added feature on model performance. This includes tactic-level analysis, particularly focusing on historically difficult-to-predict classes, such as *Initial Access*, *Impact*, *Collection* and *Reconnaissance* [15].
- **Comprehensive Evaluation of Text-Only and Multi-Modal Paradigms:** We provide a comprehensive analysis of the Text-Only and Multi-Modal approaches using a consistent dataset and evaluation methodology. This analysis offers insights into the strengths and limitations of each modelling paradigm.
- **Systematic Feature Importance and Model Interpretability Analysis:** We conduct an in-depth SHAP-inspired global analysis of both the text-Only and multi-modal supervised models to quantify the contributions of textual, numeric and categorical features across both approaches. This analysis demonstrates how structured metadata in the form of numeric and categorical features complements natural language, especially for minority and semantically ambiguous ATT&CK tactics.
- **Multi-Modal Architecture for Structured Vulnerability Metadata Integration:** We propose a multi-modal architecture for the SecRoBERTa Model that explicitly separates the use of Textual Features from both numeric and categorical features. This separation enables a better utilization of their strong auxiliary signals during the mapping process.
- **Temporal Robustness Evaluation under Syntactic Drift:** We introduce a temporal evaluation using a post-2022 test split to assess model robustness under evolving CVE descriptions. Results provide preliminary evidence that the proposed multi-modal architecture is more resilient to syntactic drift than text-only baselines.
- **Hyper-parameter Optimization with Optuna:** We apply *Optuna* [26] to fine-tune the hyper-parameters of both text-Only and multi-modal models. This optimization leads to significant performance improvements for both approaches, underscoring the importance of hyper-parameter optimization in supervised models.
- **Public Release of Resources:** All datasets, code, and supplementary materials are made publicly available via the project's GitHub repository to support reproducibility. We present and release an extended dataset comprising 7,328 CVE entries, each enriched with CWE, CPE, and CVSS information, and optionally annotated with CAPEC and EPSS data.

The rest of the paper is structured as follows: Section II reviews related work on supervised and unsupervised approaches for mapping CVEs to MITRE ATT&CK tactics and techniques. Section III outlines the methodology, including data collection, preprocessing, model development, and evaluation. Section IV presents the research objectives and hypotheses, as well as the experimental results. In Section V, we interpret the research findings and discuss their implications. Section VI provides our first round of SHAP-inspired global analysis of the text-Only model, focusing on feature importance, along with considerations for data preprocessing in preparation for a multi-modal architectural shift. Section VII demonstrates the multi-modal architectural diagram, detailing the structured metadata inputs for the approach. This section also includes our second round of feature importance evaluation and hyper-parameter retuning. In Section VIII, we address the problem of syntactic drift over time, implementing a yearly time-series split of training, validation and test data. This allows us to conduct an exploratory analysis of the robustness of both text-Only and multi-modal architectures in a dynamic, real-world context. Section IX discusses the limitations of the study, while Section X concludes with a summary of our findings and directions for future work.

II. Related work

Although prior studies have leveraged alternative data sources for MITRE ATT&CK mapping, including threat reports [27][28], malware [29][30], network traffic [31], and provenance graphs [32], this review specifically focuses on supervised and unsupervised CVE-to-TTP mapping approaches, as these are most relevant to the scope of the present research.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A. Supervised Mapping

Existing approaches to mapping CVEs to the MITRE ATT&CK framework predominantly rely on supervised learning. Branescu et al. [15] modelled this as a multi-label classification task using CVE descriptions for ATT&CK tactics mappings. Ampel et al. [16] employed self-distillation to capture long-term textual dependencies. Vulcan Cyber [18] proposed enriching input features with CWE and CVSS Version 3.x data. BERT-based models, such as CVE2ATT&CK [12], showed promise using only CVE descriptions for mapping 31 of 92 ATT&CK techniques. Mendsaikhan et al. [17] expanded coverage to 52 techniques using textual features from CVE descriptions, though performance declined with label expansion due to limited training data. Adam et al. [19] introduced a two-step mapping via CWEs, but their method is constrained by incomplete CWE annotations in CVEs and the lack of comprehensive CWE-to-ATT&CK mappings.

B. Unsupervised Mapping

The SMET framework [14] employs semantic role labelling to rank ATT&CK techniques based on CVE descriptions, without requiring labelled data. Kuppa et al. [13] demonstrated an unsupervised approach that maps CVEs to 37 ATT&CK techniques by extracting relevant phrases from both threat reports and ATT&CK descriptions. However, they observed that many CVE entries contain minimal textual content, resulting in incomplete or failed mappings.

Since SMET does not leverage structured NVD attributes (e.g., CVSS, CWE, CPE) or EPSS probability scores, it struggles with sparse or ambiguous descriptions. Furthermore, as a fully unsupervised pipeline with no learnable parameters, SMET lacks adaptability to evolving CVE patterns or domain-specific requirements. Its logistic regression classifier and embedding model are trained on ATT&CK descriptions, not CVE text, rendering the system insensitive to changes in vulnerability language, emerging exploit types, or shifts in reporting conventions.

The MITRE ATT&CK Enterprise framework comprises 14 tactics, 211 techniques, and 468 sub-techniques, making comprehensive CVE-to-technique mapping a complex task. Due to limited annotated data for many techniques, prior work has focused on a subset of them. In this study, we shift the focus to mapping CVEs to ATT&CK tactics, emphasizing higher-level adversary objectives, such as *Reconnaissance, Initial Access, Collection*, and *Impact*, all of which have been historically difficult for state-of-the-art (SOTA) methods [15].

Mapping at the tactic level offers practical benefits for vulnerability prioritization, attacker path modeling, and risk propagation [33]. To address the challenges of sparse textual descriptions and limited adaptability, we enhance both supervised and unsupervised approaches by incorporating structured NVD data (CVSS, CWE, CAPEC, CPE) and EPSS scores, going beyond CVE descriptions alone.

While our long-term goal remains mapping to techniques, we argue that tactic-level mapping can be substantially improved with richer input features. This work lays a scalable foundation for future expansion to technique-level mappings with broader dataset coverage.

Table I presents a comparative synthesis of existing work most relevant to the proposed study. Each surveyed approach is evaluated along several dimensions: the source and type of data leveraged, whether the methodology targets technique- or tactic-level mapping, the breadth of ATT&CK coverage, the learning paradigm employed, the underlying methodological framework, dataset scale, and the extent to which the solution addresses explainability, multimodal learning, and syntactic drift in vulnerability metadata over time.

As shown in the table, although prior contributions have meaningfully advanced the automation of CVE-to-TTP mapping, a consistent limitation across all surveyed works is the absence of supplementary contextual information to enrich vulnerability metadata. This constraint has precluded the adoption of multimodal learning strategies, which would otherwise enable models to exploit complementary signals across heterogeneous data sources. Without exception, existing approaches confine themselves to the textual content of CVE reports, forgoing the representational advantages that multimodal fusion may afford. Moreover, no surveyed method accounts for syntactic drift, the gradual evolution of linguistic conventions and terminology within vulnerability descriptions over time, a phenomenon that poses a substantive challenge to the temporal robustness of learned representations.

III. Methodology

The proposed methodology for automatically mapping CVEs to MITRE ATT&CK tactics comprises four main phases: dataset collection, dataset processing, mapping using both unsupervised and supervised approaches, and performance evaluation. The overall architecture is shown in Figure 1, with each phase described in detail below.

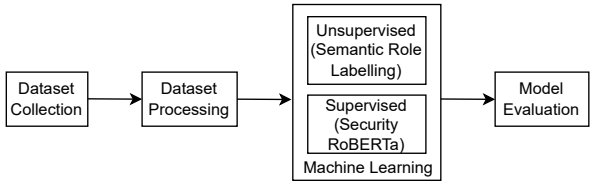


Figure 1. An overview of the proposed framework for automated mapping of CVEs to MITRE ATT&CK Tactics

A. Dataset Collection

The phase begins with the collection of an initial dataset to support both unsupervised and supervised approaches. We use datasets from SMET [14] and Branescu et al.[15], representing state-of-the-art methods in each category. These datasets include CVE IDs and their corresponding descriptions as primary features. To enrich this initial dataset, we incorporated five additional sources from the NVD and related repositories:

- CVE Descriptions (baseline): Textual descriptions of vulnerabilities from CVE entries. For instance, the CVE description for CVE-2025-49163 is "*Arris VIP1113 devices*

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. COMPARISON WITH THE STATE-OF-THE-ART METHODS. ● = FULLY PRESENT; ○ = ABSENT

Ref	Features	Mapping	Learning Paradigm	Methodology	Dataset	Explainability	Multimodal Learning	Temporal Analysis
[14]	CVE description	41 techniques	Unsupervised	Semantic Role Labelling, ATT&CK BERT, Logistic Regression Models	303 CVEs	○	○	○
[12]	CVE description	31 techniques	Supervised	Naive Bayes, CVC, CNN, BERT,SciBERT, SecBERT	1813 CVEs	○	○	○
[13]	CVE reports	17 techniques	Unsupervised	Multi-Head Joint Embedding Neural Network Model	101,410 reports	○	○	○
[16]	CVE description	10 tactics	Supervised	Self-knowledge Distillation, RoBERTa	24,863 CVEs	○	○	○
[17]	CVE description	52 techniques	Supervised	MlkNN, LabelPowerSet, ClassifierChain, RAkELd, BinaryRelevance, MlkNN	8077 CVEs	○	○	○
[15]	CVE description	14 tactics	Supervised	SecRoBERTa, SecBERT, CyBERT, TARS, GPT-4	9986 CVEs	○	○	○
Ours	CVE description, CWE, CVSS, CPE, EPSS, CAPEC	14 tactics	Unsupervised, Supervised	Semantic Role Labelling, SecRoBERTa	7328 CVEs	●	●	●

through 2025-05-30 with KreaTV SDK allow booting an arbitrary image via a crafted /usr/bin/gunzip file."

- Common Weakness Enumeration (CWE) [34]: Standardized identifiers for software weaknesses (e.g., CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')).
- Common Vulnerability Scoring System (CVSS) [35]: Quantitative scores (0–10) reflecting exploitability and impact.
- Common Platform Enumeration (CPE) [36]: Machine-readable identifiers for vulnerable software/hardware (e.g., `cpe:2.3:a:microsoft:edge:`)
- Exploit Prediction Scoring System (EPSS)[37]: Probabilities (0–1) estimating the likelihood of a CVE being exploited in the wild.
- Common Attack Pattern Enumeration and Classification (CAPEC) [38]: Titles describing adversary tactics (e.g., CAPEC-209: XSS Using MIME Type Mismatch).

NVD data was retrieved using the official API [39], EPSS scores via the EPSS API [40], and CAPEC titles were obtained through web scraping and HTML parsing of the CAPEC website.

Following enrichment, the unsupervised dataset retains its original size but is enhanced with additional features, including CWE, CPE, and CVSS data. The supervised dataset initially comprises 9,986 CVEs from prior work [15], but is reduced to 7,328 entries after filtering out CVEs lacking sufficient NVD attributes (CWE, CPE, CVSS). Features with missing values across any dataset entries are removed to prevent negative impacts on machine learning performance. This preprocessing step of filtering out rows with null values to improve system effectiveness is consistent with established machine-learning approaches for automated CVE-to-MITRE ATT&CK tactic mapping [41].

For supervised learning, we adopt an 80/20 train-test split as recommended by [15], using 80% of the 7,328 CVEs for training and 20% for testing. The *extended* dataset is enriched with EPSS scores, processed CVSS (v2/v3), CWE, CPE, and later CAPEC features. A comparative summary between the initial dataset [15] and the enriched version used in this work is provided in Table II.

TABLE II. COMPARISON BETWEEN THE INITIAL DATASET INTRODUCED BY BRANESCU ET AL.[15] AND OUR FULLY ENRICHED DATASET

ATT&CK Tactic Class	CVE Record Count	
	Initial Dataset[15]	Our Dataset
Reconnaissance	170	141
Resource Development	170	117
Initial Access	722	573
Execution	2642	1183
Persistence	3016	1591
Privilege Escalation	3218	1731
Defense Evasion	7552	5354
Credential Access	614	534
Discovery	2369	1959
Lateral Movement	1932	620
Collection	663	576
Command & Control	427	382
Exfiltration	171	126
Impact	349	286
Total	**9,986 CVEs**	**7,328 CVEs**

The unsupervised dataset comprises 827 CVEs distributed over 120 ATT&CK techniques, selected from publicly available CVE-to-ATT&CK mappings provided by the Center for Threat-Informed Defense [25], allowing for direct evaluation against verified mappings. This differs from the dataset used in SMET [14], which lacked consistent NVD feature coverage. Accordingly, the SMET baseline was re-evaluated on our 827-entry *description-only* dataset, compared to the original 303 entries used in SMET distributed over 41 techniques from the ATT&CK matrix.

Two primary datasets are created for the unsupervised and supervised mappings: (1) the *Description-Only* dataset, containing CVE IDs and textual descriptions, and (2) the *Extended* dataset, which builds upon the former by incorporating preprocessed CWE, CVSS, CPE, EPSS scores, and optionally CAPEC data.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

B. Dataset Processing

A key design decision in this research was to enrich the original CVE dataset with additional structured fields, CWE, CVSS, CPE, and optionally EPSS and CAPEC, to improve the accuracy of CVE-to-ATT&CK tactic mappings. This enrichment supports both the unsupervised (SMET) and supervised (SecRoBERTa) approaches by enhancing the semantic and contextual representation of each CVE. To ensure consistency across data types, all extended features (except EPSS scores, which are numeric) were pre-processed into natural language format. This was necessary due to the inconsistent quality of CVE descriptions and the structured, non-linguistic format of most added fields.

1) CVSS Pre-Processing: CVSS Version 2.0 and CVSS Version 3.x vector strings were transformed into natural language using mappings from the NVD CVSS calculators [42][43]. For example, the CVSS v3.1 vector `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H` of CVE-2023-23333 is pre-processed into:

- *"The CVE is Exploited by the Network Attack Vector. The CVE has Low Attack Complexity. The CVE Requires No Privileges. The CVE Does Not Require User Interaction. The CVE scope is Unchanged. The CVE has a High Confidentiality Impact. The CVE has High Integrity Impact. The CVE has High Availability Impact."*

2) CWE Pre-Processing: CWEs were converted into natural language by combining their titles and descriptions. For instance, CWE-78 is rendered as:

- *"The CVE is affected by Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection'): The product constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component."*

For multiple CWEs associated with a single CVE ID, the processed CWE strings will be concatenated to create a longer sentence.

3) CAPEC Pre-processing: CAPEC titles were processed similarly to CWEs. However, due to limited API support and outdated data (last reviewed in 2023), CAPEC enrichment was used selectively. The CAPEC feature was inadequately represented due to incomplete web scraping, which extracted only the titles of CAPEC. This limitation diminished the usefulness of the CAPEC data. A pilot study indicated that the inclusion of CAPEC features did not improve mapping accuracy.

4) CPE Pre-Processing: CPE strings were parsed to extract three key attributes: component type (application, Operating System, or hardware), vendor, and product. These were reformatted into natural language. For example: `cpe:2.3:o:contec:solarview_compact_firmware:*:*:*:*:*:*:*:*:*` was converted to: *The "CVE affects Contec Solarview_compact_firmware Operating System."*.

To reduce noise from highly variable product names, a standardization step was applied. Generic terms, such as "Product", were substituted when the product name was not essential. In contrast, critical operating system identifiers (e.g., Windows, Linux, Mac_os_x and Linux_kernel) were preserved. For instance: `cpe:2.3:a:microsoft:365_apps:-:*:*:*:enterprise:*:*:*` was converted to: *"The CVE affects Microsoft Product Application."*. This generalization improves model robustness by minimizing irrelevant variance while preserving key distinctions necessary for accurate CVE-to-ATT&CK tactic mapping.

C. Machine Learning

This research addresses the challenge of automatically mapping CVEs to the MITRE ATT&CK framework using both unsupervised and supervised machine learning techniques.

For the unsupervised approach, we employ the SMET framework [14], a state-of-the-art method that does not require labelled data. SMET extracts semantically meaningful attack vectors from CVE textual descriptions by leveraging semantic role labelling and other semantic similarity techniques. We extend the original SMET implementation, designed for *description-only* inputs, to incorporate structured features from the NVD, including CVE ID, CWE, CVSS, and CPE. We hypothesise that even in the absence of labelled data, incorporating this extended feature set enhances the quality of semantic mappings.

For the supervised approach, we utilize SecRoBERTa [44], a transformer-based model derived from RoBERTa [45], which is an optimized version of BERT (Bidirectional Encoder Representations from Transformers), and has been fine-tuned on cybersecurity-specific corpora. Prior work has shown that SecRoBERTa achieves state-of-the-art performance in mapping CVEs to ATT&CK techniques [15]. Castano et al. [46] trained five BERT-based models and found SecRoBERTa to be the most effective at linking CTI sources via external references, resulting in more complete datasets and improved threat intelligence.. We further fine-tune a pre-trained ATT&CK-BERT model from *Hugging Face* [44] using our extended dataset, which includes NVD features, EPSS probability scores, and CAPEC identifiers. Tokenization and model management are performed using the *Hugging Face* Transformers library [47]. While we maintain the default settings for batch size and number of epochs, we adjust the learning rate to `3.884755049077609e-05` and the dropout rate to `0.4864913766068174` to optimize model performance.

The dual-method study aims to investigate whether both unsupervised and supervised models can benefit from enhanced CVE representations, which could improve automated mappings to adversarial tactics in ATT&CK.

D. Model Evaluation

We evaluated the machine learning models using *accuracy, validation loss*, and both *macro* and *weighted F1 scores*.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Validation loss serves as an indicator of generalization performance, with lower values suggesting reduced overfitting or underfitting. *Accuracy* reflects the overall proportion of correctly predicted tactic labels. The *macro F1 score*, as the unweighted average of per-tactic F1 scores, emphasizes performance on less frequent classes. The *weighted F1 score*, our primary metric, accounts for class imbalance by weighting each tactic's F1 score by its frequency.

IV. EVALUATION

To support a comprehensive evaluation of our system, we define specific hypotheses for validation:

- **H1**: An *extended* dataset improves overall mapping accuracy compared to the commonly used *description-only* datasets [12][14][15][17].
- **H2**: Tactics that are typically harder to classify, such as *Reconnaissance, Initial Access, Collection*, and *Impact*, as identified by Branescu et al. [15], will show improvements in their F1-scores.
- **H3**: Hyperparameter tuning leads to additional gains in mapping accuracy.

A. Unsupervised Mapping Validation

Given that SMET is an unsupervised methodology that ranks mappings based on semantic similarity. In the proposed solution, rankings greater than 0.1 are considered potentially correct mappings. When an entry is labelled as *Completely Accurate*, it indicates a 1:1 match with the testing data provided for a CVE. If the entry is designated as *Semi-Accurate*, it means that while the accepted mappings included correct ones, they also incorporated some incorrect mappings that exceeded the threshold. Conversely, if the entry is marked as *Inaccurate*, it signifies that no correct mappings were obtained that met the threshold (>0.1). We compared the SMET results from a *Description Only* dataset with an *extended* dataset, verifying an increase in mapping accuracy. As shown in Table III, the enriched unsupervised dataset with CVSS vectors, CWE and CPE summaries outperformed the baseline *description only* in every metric.

TABLE III. COMPARISON OF UNSUPERVISED MAPPING ACCURACY ON 828 CVES, *Description Only* VS. *Enriched* DATASET

	Description Only	**Enriched (+CVE, CVSS & CPE)**	**Impact (%)**
Completely Accurate	6	10	+66.7
Semi Accurate	88	125	+42.0
Inaccurate	733	692	-5.6

Despite notable improvements in mapping accuracy, over 80% of CVEs remain incorrectly mapped without supervised learning, demonstrating that feature enrichment alone is insufficient. While enriched features capture semantically meaningful attack vectors, they do not match the performance of supervised models. Results show that mapping CVEs to ATT&CK Techniques suffers from low accuracy due to limited labelled data.

For example, SMET, a state-of-the-art unsupervised method, uses text similarity between CVEs and ATT&CK technique descriptions, enabling semantic mapping of 303 CVEs to 41 techniques. In contrast, our improved unsupervised dataset includes 828 CVEs mapped to 120 techniques, with both datasets averaging 7 CVEs per technique. Meanwhile, the leading supervised dataset includes 9,985 CVEs across 14 tactics, with each tactic supported by a minimum of 170 samples and an average of 713 entries (see Table II). This data imbalance leads to better performance when mapping to ATT&CK Tactics rather than Techniques.

Given the limited performance of unsupervised methods, this research adopts a supervised approach. Nonetheless, the unsupervised results confirm that *enriched* datasets are more effective than *description-only* inputs for offensive technique mapping.

B. Supervised Mapping Validation

This section presents the results of the supervised learning experiments. First, we analyze performance across dataset variants to assess the impact of added features, including comparisons with and without CAPEC. Second, we report per-tactic F1 scores for MITRE ATT&CK tactics. Finally, we benchmark our approach against the state-of-the-art method by Branescu et al. [15].

Table IV presents the overall performance across the supervised dataset variants, enabling a detailed comparison of feature-specific contributions. Incorporating the EPSS feature alone consistently improves all four performance metrics: validation loss, accuracy, macro F1 score, and weighted F1 score, relative to the *description-only* baseline. Similar improvements are observed when CWE, CPE, and CVSS features are added, each contributing to increased model performance. In contrast, the inclusion of the CAPEC feature results in a decline across all four metrics, with the CAPEC Title-only extension causing a particularly notable degradation. As a result, CAPEC was excluded from the fully extended feature set. All other feature combinations outperform the description-only baseline, thereby supporting Hypothesis **H1**. Additionally, hyperparameter tuning yields a consistent performance boost over the enriched but untuned models, with gains of approximately 2% to 3% in mapping accuracy and macro F1 score, thereby supporting Hypothesis **H3**.

Table V presents F1 scores for the *description-only* baseline, the fully enriched dataset (with and without tuning), and the Branescu et al. [15] SecRoBERTa model. The most notable improvements were observed for hard-to-predict tactics: *Initial Access* improved from 65.27% to 67.44%, *Collection* from 79.44% to 84.34%, *Impact* from 67.57% to 72.00%, and *Reconnaissance* from 37.33% to 46.15%, confirming Hypothesis **H2**. Medium-difficulty tactics, such as *Credential Access* and *Command & Control*, saw moderate gains of 1%–7%. Well-predicted tactics, such as *Defense Evasion, Discovery,*

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE IV. OVERALL PERFORMANCE ACROSS SUPERVISED DATASET VARIANTS.

Supervised Dataset Variant	Validation Loss	Accuracy	Macro F1 Score	Weighted F1 Score
Description Only	0.0747	0.8286	0.7948	0.9232
Description + EPSS	0.0729	0.8335	0.8138	0.9277
Description + CWE	0.0724	0.8407	0.7979	0.9248
Description + CVSS	0.0815	0.8229	0.8024	0.9163
Description + CPE	0.0746	0.8286	0.8050	0.9244
Description + CAPEC	0.0870	0.8179	0.7119	0.9011
Fully Extended (Description + EPSS + CVSS + CPE)	0.0743	0.8383	0.8144	0.9245
Fully Extended + Tuned	**0.0658**	**0.8538**	**0.8401**	**0.9347**

Privilege Escalation, Persistence, Lateral Movement, and *Execution* began above 90% and saw only marginal improvements (1%–2%), with fine-tuning contributing an additional 0.5%–1%. The final model achieved a 93.5% weighted F1 score.

TABLE V. PLOTS PER-CLASS F1 SCORES FOR THE SUPERVISED DATASET VARIANT AND COMPARISON BETWEEN THE SECROBERTA PER-CLASS ON DESCRIPTION ONLY REPORTED IN [15]

Tactics	Description only (Benchmark)	Full Extended Dataset (+EPSS+CWE +CVSS+CPE)	+ Optuna Fine-Tuning	SOTA [15]
Reconnaissance	37.33%	36.73%	**46.15%**	53.84%
Resource Development	51.47%	65.81%	**65.79%**	79.13%
Initial Access	65.27%	61.52%	**67.44%**	37.18%
Execution	89.56%	89.25%	**89.95%**	74.43%
Persistence	94.42%	94.52%	**94.87%**	80.78%
Privilege Escalation	94.66%	94.90%	**95.11%**	80.46%
Defense Evasion	98.67%	98.00%	**98.41%**	91.96%
Credential Access	84.82%	89.39%	**91.81%**	67.27%
Discovery	97.24%	97.29%	**97.92%**	81.55%
Lateral Movement	92.87%	94.59%	**94.97%**	81.37%
Collection	79.44%	81.82%	**84.34%**	51.47%
Command & Control	95.21%	95.81%	**96.43%**	61.79%
Exfiltration	64.23%	74.83%	**81.01%**	88.88%
Impact	67.57%	65.73%	**72.00%**	31.11%

Compared to *Branescu et al.'s*[15] model trained on 9,986 CVEs (weighted F1: 78.88%), our approach, applied to 7,786 CVEs, achieves 93.45%. This performance gain is attributed to structured feature enrichment from NVD (CWE, CVSS, CPE) and the addition of EPSS, coupled with effective hyperparameter tuning. The results demonstrate that enriched features and tuning significantly enhance CVE-to-ATT&CK tactic mapping accuracy, especially for previously underperforming tactics.

To ensure a fair comparison with Branescu et al. [15], we grouped predicted MITRE ATT&CK tactics into three difficulty levels based on F1 scores: *hard* (<60%), *medium* (60%–80%), and *easy* (>80%). As shown in Table VI, Branescu et al. [15] identified *four* hard, *four* medium, and *six* easy tactics. Using our enriched dataset and improved processing pipeline, our method reduced the number of hard tactics to *one*, with *three* medium and *ten* easy-to-predict tactics. Notably, *three* tactics previously classified as hard in [15] were reclassified as two medium and one easy. In comparison, three of the four medium tactics were reclassified as easy in our mapping methodology. The *six* easy tactics remained unchanged. These results support Hypothesis **H2**, demonstrating that dataset enrichment and enhanced modelling reduce classification difficulty for previously challenging tactics. This finding is consistent with the results reported by *Mimoto et al.*[48], which demonstrate that incorporating supplementary contextual information, such as CWE and CAPEC's description and mitigation into vulnerability embeddings, enhances the ability to establish links between vulnerabilities and associated techniques and TTP chains.

TABLE VI. COMPARISON BETWEEN THE PREDICTED ATT&CK TACTICS BY BRANESCU ET AL. [15] AND OUR PROPOSED APPROACH

Difficulty Level	ATT& CK Tactics in Branescu et al. [15]	ATT&CK Tactics in our proposed approach
Hard	Reconnaissance, Collection, Initial Access, Impact	Reconnaissance
Medium	Resource Development, Credential Access, Execution, Command & Control	Resource Development, Initial Access, Impact
Easy	Privilege Escalation, Discovery, Persistence, Exfiltration, Defense Evasion, Lateral Movement	Privilege Escalation, Discovery, Persistence, Exfiltration, Defense Evasion, Lateral Movement, Execution, Credential Access, Collection, Command & Control

V. DISCUSSION

According to the CVE Key Details Phrasing Guidelines by MITRE [49], a comprehensive CVE description should articulate several key aspects, including the vulnerability type or root cause, attack vector, impact, attacker type, component identification, affected product(s) and version(s), and product vendor(s). However, despite the importance of these details, many CVE descriptions and their associated references suffer from inconsistencies, a lack of structure, or insufficient information [18][23][50], which poses a significant challenge for downstream tasks such as mapping vulnerabilities to offensive tactics and techniques, particularly within the MITRE ATT&CK framework.

This research demonstrates that supplementing CVE descriptions with structured data, such as software weaknesses (CWE), platform identifiers (CPE), exploit prediction scores (EPSS), attack patterns (CAPEC), and vulnerability scoring metrics (CVSS vector strings), can significantly enhance the completeness and utility of CVE records. By enriching the original textual descriptions with these standardized attributes, the proposed approach improves the effectiveness of both supervised and unsupervised models for mapping CVEs to ATT&CK techniques.

Vulnerability descriptions serve as a critical foundation in the identification and communication of security weaknesses in software, systems, and hardware. High-quality descriptions not only support threat assessment and mitigation but are also essential for enabling automated systems to aid in vulnerability

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

prioritization and response. This study contributes to the growing body of work aimed at automating the mapping of CVEs to adversary behaviour models, thereby advancing vulnerability analysis and threat-informed defence.

VI. Extended Feature Analysis of our Current String Concatenated Model

A. SHAP Feature Importance

To better understand how the baseline string-concatenated model leverages the additional features introduced in our dataset, we conduct a global feature importance analysis inspired by SHAP (SHapley Additive exPlanations) [51]. SHAP has been widely adopted in machine learning (ML) for cybersecurity to interpret, explain, and validate the decisions of complex "black-box" models, with prominent applications in cyber attack detection and defense [52]. Whereas prior studies primarily employ SHAP to justify or validate predictive outputs, our objective is to examine the internal reasoning patterns of the model.

More specifically, we use SHAP-inspired attribution to investigate why a particular vulnerability description is mapped to a given tactic, rather than merely confirming that such a mapping occurs. The purpose of this analysis is not to improve predictive performance, but to quantify the relative contribution of each feature group to the model's tactic predictions and to determine whether non-textual vulnerability metadata is being meaningfully incorporated into the decision-making process.

1) Analysis Methodology: In the current architecture, no explicit distinction is made between numeric and textual features. All inputs are concatenated into a single textual sequence, tokenized, and processed uniformly by the transformer encoder. Consequently, numeric attributes are represented solely through token-based encoding, raising concerns that their semantic meaning may be diluted or inadequately captured when treated as plain text rather than structured numerical inputs.

To assess the contribution of the additional features incorporated in the concatenated-input model, we conduct a SHAP-inspired feature importance analysis based on attention-derived attribution. Specifically, feature importance is approximated using attention weights from the final transformer layer. These weights are aggregated from the `[CLS]` token to the corresponding input tokens and subsequently grouped at the feature-category level. The resulting importance scores are reported for the following feature groups: Abstract Text (CVE description), EPSS, CVSS, CWE description, and CPE.

For empirical evaluation, we sample 200 CVEs from the training/validation split and aggregate feature importance scores with respect to the predicted tactics across this subset. Although this sample represents approximately 3% of the full dataset, it provides a manageable scope for detailed analysis within the constraints of this study. While this method does not compute exact Shapley values, attention-based attribution offers a computationally efficient and internally consistent proxy for estimating global feature importance, thereby facilitating comparative analysis across different architectural configurations

2) Global Analysis Results: Our global analysis reveals a clear dominance of natural language features. Specifically, the Abstract Text (CVE descriptions) consistently accounts for approximately 40–60% of total feature importance across all tactics, irrespective of class frequency, as shown in Table VII. This observation is consistent with prior research in vulnerability analysis, which demonstrates that CVE descriptions alone often provide sufficient information for automated vulnerability classification, adversarial technique mapping, and severity prediction tasks [12][53][54].

In addition, CWE emerges as the second most influential feature group, contributing approximately 24–35% of overall importance. This suggests that the categorical and repetitive structure of CWE representations, introduced during the dataset pre-processing stage, may align more effectively with token-based transformer models. This finding is also consistent with broader research in software vulnerability management, particularly in vulnerability exploitation prediction. Prior studies indicate that although CVE descriptions account for the majority of predictive performance, the inclusion of structured auxiliary features can provide incremental improvements without fundamentally altering overall performance patterns [55].

In contrast, other security-relevant features remain comparatively underutilized. EPSS contributes less than 2% across all tactics, while CVSS accounts for approximately 5–7%, as shown in Table VIII. Similarly, CPE exhibits limited influence, ranging from approximately 1% to at most 8% feature importance.

3) Interpreting the Results and Limitations of the Concatenated String approach: Transformer-based models such as SecRoBERTa demonstrate strong performance when processing semantically rich textual inputs. However, structured numeric features, such as CVSS and EPSS, appear to be underrepresented within this framework due to their numerical nature. When converted into textual tokens, these features lose their quantitative semantics, potentially resulting in representational inefficiencies and limiting their effective contribution to the model's predictive capacity.

Given that the concatenated-input model systematically underutilizes structured metadata, we propose transitioning to a multimodal architecture. This approach integrates numeric features through dedicated embedding representations while retaining a concatenated textual sequence for semantically meaningful inputs (i.e., Abstract Text, CWE, and CPE). Such a design aims to preserve the inherent structure of numerical attributes while leveraging the transformer's strengths in contextual language understanding.

B. Data Pre-processing Considerations

Given that *Abstract Text* and *CWE* emerged as the dominant contributors in our prior global analysis, we retain their existing representation, as both provide semantically rich information that is well-suited to transformer-based encoding.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE VII. DOMINANT FEATURE IMPORTANCE (%) BY TACTIC FOR THE STRING-CONCATENATED BASE MODEL.

Tactic	Abstract	CWE	Samples
Reconnaissance	62.75	33.89	2
Resource Development	40.71	46.65	1
Initial Access	66.05	27.91	14
Execution	63.03	28.20	38
Persistence	57.55	32.71	44
Privilege Escalation	58.12	31.89	51
Defense Evasion	63.29	24.29	143
Credential Access	54.43	35.14	15
Discovery	64.90	23.71	57
Lateral Movement	60.99	29.06	17
Collection	57.16	32.62	10
Command and Control	57.31	29.70	7
Impact	57.63	31.19	6

N.B. Importance values may not sum to exactly 100% due to rounding. Samples indicate the number of instances contributing to the global SHAP analysis for each tactic.

TABLE VIII. UNDER-UTILISED FEATURE IMPORTANCE (%) BY TACTIC FOR THE STRING-CONCATENATED BASE MODEL.

Tactic	CPE	CVSS	EPSS
Reconnaissance	1.13	1.69	0.53
Resource Development	8.79	3.45	0.40
Initial Access	1.83	3.50	0.71
Execution	3.31	4.14	1.32
Persistence	3.79	4.80	1.15
Privilege Escalation	3.78	4.90	1.31
Defence Evasion	3.71	7.23	1.49
Credential Access	4.27	5.05	1.11
Discovery	3.90	6.44	1.06
Lateral Movement	3.99	4.50	1.46
Collection	3.36	5.88	0.98
Command and Control	4.16	7.54	1.29
Impact	4.86	4.67	1.66

N.B. Importance values may not sum to exactly 100% due to rounding.

In contrast, the textual encoding of numeric features removes their inherent inductive bias and structured semantics within the classification process. Accordingly, our architectural modification prioritizes improving the representation and integration of the previously underutilized feature groups, namely CPE, CVSS, and EPSS, to enhance their effective contribution to tactic prediction.

1) CVSS Pre-Processing for Multi-Modal Shift: CVSS scores constitute inherently structured and ordinal measures of vulnerability severity and are widely employed by security practitioners for risk assessment and prioritization. In our earlier preprocessing pipeline (see Section III-B), CVSS vectors were converted into natural language strings, thereby discarding their explicit ordinal structure and the objective severity relationships encoded within individual vector components. This transformation obscures magnitude relationships and may hinder effective gradient-based learning, as the model can no longer directly exploit the structured numeric properties of the feature.

To address this limitation, we propose preserving the numerical integrity of the CVSS representation by decomposing the vector string into its constituent components. Specifically, we separate Continuous Impact metrics (Confidentiality, Integrity, and Availability: C, I, A) from Categorical Exploitability metrics (Attack Vector, Attack Complexity, Privileges Required, User Interaction, and Scope: AV, AC, PR, UI, Scope). Continuous impact scores are maintained as numerical inputs to preserve magnitude relationships, while categorical exploitability metrics are embedded to capture configuration-dependent effects.

This separation of concerns in encoding the CVSS vector facilitates the learning of meaningful interactions between severity and exploitability dimensions. Moreover, it enables the model to distinguish between what is impacted (impact metrics) and how the vulnerability can be exploited (exploitability metrics), thereby preserving the structural semantics inherent in the CVSS framework.

2) EPSS Pre-Processing for Multi-Modal Shift: EPSS represents the probability of real-world exploitation as a calibrated probabilistic score [56]. Encoding this value as textual tokens obscures its monotonic structure and undermines its probabilistic interpretation, thereby limiting the model's ability to leverage its quantitative semantics. To preserve these properties, we retain EPSS as a continuous scalar feature and incorporate it through a dedicated numeric embedding.

Although prior analysis indicated relatively low feature importance for EPSS, we maintain its inclusion due to its complementary relationship with CVSS. Whereas CVSS captures impact severity, EPSS reflects exploitation likelihood; together, they provide orthogonal dimensions of vulnerability risk. Preserving EPSS in its native numeric form maintains this conceptual distinction and enables the model to learn interactions between severity and probability without compromising their respective semantic structures.

3) CPE Dual-Representation for Multi-Modal Shift: As discussed in Section III-B, which outlines the preprocessing strategy for CPE in the string-concatenated model, this feature comprises both descriptive natural language identifiers and structured categorical information (e.g., system type). Representing CPE using a single encoding modality would therefore risk discarding valuable structural or semantic information.

In the proposed architectural revision, we decompose CPE into two components: (i) a categorical system-type attribute (operating system, application, or hardware), and (ii) a cleaned textual description, following the preprocessing procedure previously described. This dual representation provides two key advantages. First, categorical encoding facilitates system-level generalization across broad platform classes. Second, textual encoding preserves vendor and product-specific details, enabling fine-grained semantic differentiation. Together, this

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

separation supports both abstraction and specificity within the learned representations.

VII. Multi-Modal Architecture

The motivation for transitioning from a string-concatenated architecture to a multi-modal framework arises from the underutilization of numeric features in the former approach. When numeric variables are converted into textual tokens, their ordinal and proportional properties are disrupted through tokenization, leading to diminished feature importance in global analyses. The objective of adopting a multi-modal architecture is not to reduce reliance on textual information, but rather to enable structured metadata to contribute to predictions in a manner that preserves their quantitative integrity. Figure 2 presents an overview of the proposed architectural transition from string-concatenated to a multi-modal framework. Each component of the architecture is discussed in subsequent sections.

A practical concern in deploying automated mapping systems for CVE to ATT&CK is the *cold-start* problem. Zero-day threats and newly published CVEs often lack complete structural metadata at the time of ingestion by the National Vulnerability Database (NVD). Consequently, a model that relies heavily on structured inputs would be of limited use in real-time production environments. The proposed multimodal architecture directly addresses this challenge with its decoupled branch design. Specifically, the text-only SecRoBERTa branch operates independently of the structured metadata branch. When structured fields are missing, common with newly disclosed vulnerabilities, the model defaults to text-only inference without needing any architectural changes or retraining. This feature represents a graceful degradation property in which, instead of failing when complete metadata is unavailable, the system uses structured signals when they exist and resorts to contextual language modelling when they do not. As a result, the model's dependency on metadata is not binary but rather gradual, maintaining predictive capacity even under conditions of sparse input.

Additionally, it's important to note that the absence of complete metadata is rarely the reality for newly published CVEs. Usually, a subset of structured fields becomes available soon after initial disclosure. For example, CVSS base scores and CWE identifiers are typically assigned within days of NVD ingestion, and EPSS scores, which quantify the likelihood of exploitability, are recalculated and published daily. The proposed architecture does not require all structured inputs to be filled out at the same time; instead, these fields can be gradually incorporated as the vulnerability record evolves. This allows prediction confidence to improve incrementally as more metadata becomes available. This staged enrichment pattern aligns well with operational Security Operations Center (SOC) workflows, where vulnerability records undergo continuous revision and enhancement following initial publication. Taken together, these characteristics suggest that the proposed framework is effective not only under controlled experimental conditions but also well-suited for the dynamic and incomplete information environments typical of real-world vulnerability management.

A. Input Modalities and Feature Encoding

1) Textual Inputs: With respect to textual inputs, the Abstract Text and CWE representations from the concatenated-text model are retained to preserve their syntactic and semantic contributions. The CPE feature is represented using a dual approach; within the textual modality, Product and Vendor information are expressed in natural language form. These string-based features are concatenated into a unified sequence and tokenized using the same preprocessing pipeline and maximum sequence length as the baseline model. Crucially, numeric and categorical structured metadata are excluded from this tokenized sequence, preventing their transformation into textual tokens and thereby allowing their effects to be evaluated independently within the multi-modal architecture.

2) Continuous Numeric Inputs: Continuous numeric features are represented as real-valued inputs rather than textual tokens, as found in the String Concatenated model. EPSS is included as a single scalar capturing probabilistic exploitation likelihood, while CVSS impact components (Confidentiality (C), Integrity (I), Availability (A)) are mapped to continuous values reflecting severity magnitude. This design preserves the ordinal and proportional relationships inherent in these metrics, relationships that are otherwise lost when such variables are embedded as text.

3) Categorical Inputs: Discrete categorical features, including CVSS exploitability metrics (Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), User Interaction (UI), Scope (S)) and CPE system type, are encoded using learned embeddings. These variables represent configuration and access constraints that do not exhibit ordinal relationships, making embedding-based representations more appropriate than scalar encodings. This approach enables the model to learn similarity structures between categorical states without imposing artificial distance assumptions.

B. Multi-Modal Fusion Architecture

The proposed multi-modal architecture utilizes a late-fusion strategy, wherein the pooled `[CLS]` representation from the SecRoBERTa encoder is concatenated with learned embeddings of structured numeric and categorical features. This combined representation is subsequently fed into a fully connected classification head to generate tactic-level predictions. Late fusion was adopted to preserve the intrinsic functionality of the text encoder while allowing structured metadata to influence downstream decision boundaries without disrupting language modeling. This approach is particularly advantageous for integrating modalities with heterogeneous feature spaces, as it enables each modality to be processed by specialized, independent architectures that extract high-level semantic representations before fusion [57].

C. Hyper-parameter Re-tuning

Given the substantial architectural differences between our string-concatenated and multi-modal approaches, hyperparam-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

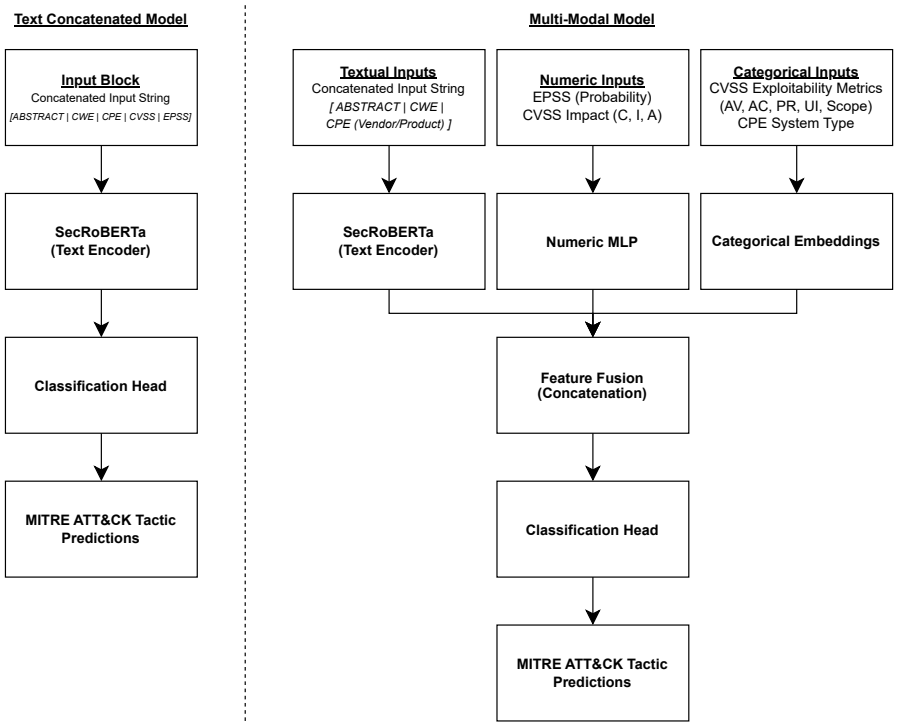


Figure 2. An overview of the proposed architectural Shift from string-concatenated to a multi-modal framework

eter retuning using *Optuna* was conducted. A constrained search space was defined to ensure a feasible and efficient fine-tuning procedure, focusing on the *learning rate*, *transformer dropout*, *numeric projection size*, and *numeric dropout*. To limit computational overhead, a fixed subsample of the training data was employed during tuning, with a weighted F1 score used as the optimization objective to maintain consistency with the established evaluation protocol. The optimal hyper-parameters identified via Optuna for the multi-modal architecture are as follows:

- Learning Rate: 5.63×10^{-5}
- Transformer Dropout: 0.175
- Numeric Hidden Size: 64
- Numeric Dropout: 0.028

D. Discussion of Multi-Modal Architecture Performance

1) Overall Performance Gain: Table IX presents a comparison between the string-concatenated and multi-modal model approaches. The results indicate that separating numeric and categorical features from the dominant textual features provides immediate benefits, even before the second round of Optuna hyper-parameter tuning. Following Optuna-based fine-tuning, the multi-modal model demonstrates substantial improvements, most notably in Macro F1, which increases from 0.8243 in the string-concatenated model to 0.8972. This suggests that incorporating structured metadata particularly enhances performance on minority and harder-to-predict tactics, rather than merely improving predictions for dominant classes. Corresponding increases in Micro F1 and Weighted F1 indicate that these gains do not compromise overall classification stability or precision. Notably, these improvements are achieved without modifying the underlying text encoder, reinforcing the conclusion that the observed performance gains arise from the integration of structured metadata within the multi-modal architecture rather than changes to the language modelling itself.

TABLE IX. OVERALL PERFORMANCE COMPARISON BETWEEN STRING-CONCATENATED AND MULTI-MODAL ARCHITECTURES

Model	Micro F1	Macro F1	Weighted F1
String-Concatenated (Best)	0.9375	0.8243	0.9347
Multi-Modal (Untuned)	0.9412	0.8382	0.9358
Multi-Modal (Tuned)	**0.9602**	**0.8972**	**0.9582**

2) Per-Tactic Behaviour: To evaluate the impact of architectural transitions on specific MITRE ATT&CK tactics, we present a comparative analysis of F1 scores in Table X. This analysis contrasts the string-concatenation baseline with both untuned and tuned multimodal variants. The results show that the fine-tuned multimodal approach yields varying performance gains across different tactics.

Notably, this architecture significantly benefits tactics characterized by high semantic ambiguity or those that have lower representation in the dataset, as established in Table II. For instance, the tactics of *Reconnaissance*, *Resource Development*, and *Exfiltration* have fewer samples compared to other

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

tactics, as highlighted in Table II. In the tuned multimodal architecture, these tactics experienced increases of 29.69%, 16.12%, and 19.28%, respectively.

In contrast, tactics with larger sample sizes, such as *Defence Evasion*, *Privilege Escalation*, and *Discovery*, displayed only modest increases of 0.91%, 1.87%, and 1.05%, respectively. Furthermore, tactics identified as *hard* to predict in Table VI include *Reconnaissance*, *Collection*, *Initial Access*, and *Impact*. However, in the fine-tuned multimodal solution, these tactics were reclassified as *medium*, *easy*, *medium*, and *easy*, respectively.

TABLE X. PER-TACTIC F1 COMPARISON BETWEEN STRING-CONCATENATED AND MULTI-MODAL ARCHITECTURES

Tactics	String-Concatenated	Multi-Modal (Untuned)	Multi-Modal (Tuned)
Reconnaissance	35.53%	36.24%	**65.22%**
Resource Development	67.50%	73.17%	**83.62%**
Initial Access	66.49%	67.18%	**79.00%**
Execution	91.15%	90.66%	**92.51%**
Persistence	94.78%	95.14%	**96.28%**
Privilege Escalation	94.97%	95.37%	**96.84%**
Defence Evasion	98.35%	98.68%	**99.26%**
Credential Access	91.74%	92.90%	**96.84%**
Discovery	97.98%	98.16%	**99.03%**
Lateral Movement	94.77%	95.54%	**96.75%**
Collection	83.03%	83.31%	**90.55%**
Command & Control	96.12%	95.93%	**97.55%**
Exfiltration	71.33%	84.52%	**90.61%**
Impact	70.27%	66.67%	**81.63%**

3) Interpretation of Performance: The performance trends observed in both the tuned and untuned multi-modal models, including both per-tactic and overall F1 scores, indicate that the addition of structured numeric and categorical metadata enhances decision-making in situations where textual descriptions are inadequate or ambiguous. While dominant tactics already achieve nearly optimal performance levels in the string-concatenated baseline model, minority tactics and those that are semantically overlapping show significantly greater improvements when using the multi-modal architecture, particularly after fine-tuning. It is important to note that, although the string-concatenation results reported in Table V indicate that the state-of-the-art (SOTA) model achieves higher per-tactic F1 scores for the *Reconnaissance* and *Resource Development* tactic classes, the proposed multimodal approach consistently outperforms the SOTA across all tactics. These results demonstrate that multimodal learning yields improved accuracy in linking vulnerability metadata to adversarial tactics. Despite the shift to a multimodal approach, the F1-score for the *Reconnaissance* tactic remains at 65.22%, which is the lowest among the 14 tactics. This score is also the lowest recorded in the text concatenated model, as shown in Table V. However, the multimodal shift resulted in a significant increase of 19.07%, improving the difficulty level of predicting Reconnaissance from hard to medium. Additionally, it is noteworthy that the per-tactic F1-score reported by *Branescu et al.* [15] was 53.84%, highlighting the improvements achieved through multimodal learning.

The substantial increase in Macro F1, in contrast to the comparatively smaller rise in Micro F1, suggests that the incorporation of structured metadata is more likely to enhance the confidence of class-level mappings rather than overall prediction certainty. This pattern is particularly evident in tactics with small sample sizes, such as *Reconnaissance*, *Exfiltration*, and *Resource Development*, which experience disproportionately larger gains compared to other tactics. In these instances, numerical and categorical features serve as auxiliary signals that help refine predictions when textual data alone is insufficient or lacks clear distinguishing patterns.

On the other hand, well-represented tactics like *Defence Evasion* show only marginal performance improvements from the addition of structured metadata features. This is because the auxiliary signals are less necessary for tactics with stable decision boundaries, thus impacting those that struggle with poor textual descriptions more significantly.

These findings lead us to conclude that the value of structured metadata in mapping CVEs to MITRE ATT&CK tactics lies not in replacing the dominance of textual information in the mapping process, but rather in providing a robust secondary signal that stabilizes decision boundaries when textual descriptions are lacking.

E. Untuned Multi-Modal Feature Importance

To evaluate the impact of the architectural shift on feature importance within the model, we conduct another round of SHAP-inspired global analysis on the Untuned Multi-Modal Model. We use the same methodology and sample selection as outlined in Section VI. This approach allows for a direct comparison of the feature importance profiles between the String Concatenated Model and the Multi-Modal Model.

In contrast to the string concatenated model, the multi-modal architecture shows a more balanced distribution between Textual and Structured Features. Although Abstract Text and CWE remain the primary contributors, along with the split representation CPE, where the text-Based component is prominently represented in certain tactics, CVSS Impact Components (aggregated for simplicity in Table XII as a single importance score), and EPSS consistently demonstrate increased importance. This confirms that the architectural separation allows numeric features to influence predictions in a meaningful way, rather than being diluted by token-based encoding.

F. Tuned Multi-Modal Feature Importance

After fine-tuning the hyperparameters, we performed a global analysis inspired by SHAP to investigate how this process affects the contribution of each modality to feature importance. The results show that while fine-tuning does not

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE XI. TEXTUAL FEATURE IMPORTANCE (%) BY ATT&CK TACTIC FOR THE UNTUNED MULTI-MODAL MODEL.

Tactic	Abstract	CWE	CPE
Reconnaissance	51.80	43.79	3.02
Resource Development	54.62	42.17	2.62
Initial Access	66.12	27.22	5.09
Execution	62.78	31.42	4.91
Persistence	55.72	38.05	5.61
Privilege Escalation	57.09	36.45	5.67
Defence Evasion	58.61	34.57	6.48
Credential Access	41.86	52.68	4.82
Discovery	60.14	31.76	7.40
Lateral Movement	59.44	34.23	5.40
Collection	63.63	30.48	4.91
Command and Control	61.91	31.77	5.84
Exfiltration	57.15	37.21	4.41
Impact	64.60	29.25	4.42

N.B. Importance values may not sum to exactly 100% due to rounding.

TABLE XII. NUMERICAL FEATURE IMPORTANCE (%) BY ATT&CK TACTIC FOR THE UNTUNED MULTI-MODAL MODEL.

Tactic	CVSS (C+I+A)	EPSS	Samples
Reconnaissance	1.30	0.09	2
Resource Development	0.54	0.06	6
Initial Access	0.77	0.80	20
Execution	0.50	0.38	45
Persistence	0.51	0.10	46
Privilege Escalation	0.64	0.15	49
Defence Evasion	0.27	0.07	161
Credential Access	0.39	0.24	16
Discovery	0.57	0.12	50
Lateral Movement	0.71	0.23	26
Collection	0.69	0.28	17
Command and Control	0.36	0.12	9
Exfiltration	1.18	0.05	1
Impact	1.24	0.49	9

N.B. Importance values may not sum to exactly 100% due to rounding. Samples indicate the number of instances contributing to the global SHAP analysis for each tactic.

TABLE XIII. TEXTUAL FEATURE IMPORTANCE (%) BY ATT&CK TACTIC FOR THE TUNED MULTI-MODAL MODEL.

Tactic	Abstract	CWE	CPE
Reconnaissance	56.97	39.07	2.47
Resource Development	55.91	40.60	2.64
Initial Access	72.71	23.90	2.48
Execution	68.09	26.94	4.24
Persistence	61.55	31.56	6.48
Privilege Escalation	61.87	31.21	6.25
Defence Evasion	63.40	29.12	7.14
Credential Access	50.81	44.70	3.63
Discovery	61.15	29.99	8.53
Lateral Movement	65.39	28.01	5.52
Collection	60.91	32.03	5.78
Command and Control	61.42	29.53	8.15
Exfiltration	59.08	37.99	2.36
Impact	64.95	30.21	3.37

N.B. Importance values may not sum to exactly 100% due to rounding.

TABLE XIV. NUMERICAL FEATURE IMPORTANCE (%) BY ATT&CK TACTIC FOR THE TUNED MULTI-MODAL MODEL.

Tactic	CVSS (C+I+A)	EPSS	Samples
Reconnaissance	1.46	0.03	2
Resource Development	0.84	0.02	6
Initial Access	0.75	0.15	19
Execution	0.61	0.12	43
Persistence	0.30	0.11	44
Privilege Escalation	0.52	0.15	46
Defence Evasion	0.22	0.12	158
Credential Access	0.75	0.10	16
Discovery	0.26	0.07	47
Lateral Movement	0.75	0.32	24
Collection	1.04	0.24	16
Command and Control	0.74	0.15	10
Exfiltration	0.36	0.21	1
Impact	1.01	0.47	9

N.B. Importance values may not sum to exactly 100% due to rounding. Samples indicate the number of instances contributing to the global SHAP analysis for each tactic.

increase the importance of structured features, it does maintain their consistent contribution across various tactics. Furthermore, fine-tuning enhances the stability of the importance of text-dominant features when sufficient semantic signals are present. This continued significance of structured data features indicates that fine-tuning improves the utilization of structured metadata rather than diminishing its influence. The persistence of these signals supports the notion that numeric features act as secondary decision modifiers.

G. Discussion on Multi-Modal Feature Importance

The analysis of feature importance in both untuned and tuned multi-modal models reveals that our architectural shift from a string-concatenated model reduces the underutilization of structured metadata. While textual features, particularly Abstract Text and CWE, remain the primary contributors to model predictions, structured and numeric inputs such as CVSS Impact Components and EPSS consistently show measurable importance when not limited to token-based encoding.

Although the importance of numeric metadata is relatively small, its contribution is both stable and semantically relevant to the associated ATT&CK Tactics. This suggests that the benefit of the multi-modal architecture arises not from replacing textual features as the main contributor, but from allowing structured metadata to provide a numerically reliable complementary signal.

Our examination of the improvements in the fine-tuned multi-modal model shows an approximate 7% increase in Macro F1 Scores. This indicates that these features particularly enhance the performance for minority and ambiguous tactics, thus improving generalization rather than merely inflating overall confidence.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

These findings support the conclusion that our multi-modal architecture more accurately reflects the true nature of CVE Data. Within this framework, natural language descriptions provide contextual understanding, while structured metrics offer insights into additional vulnerability knowledge, including severity, exploitability, and system-level constraints.

In comparison to the untuned multi-modal model, fine-tuning mainly enhances textual dominance in areas with strong semantic signals (e.g., *Initial Access* and *Execution*), while keeping numeric contributions stable. This indicates improved confidence calibration without a shift in reliance on modalities.

VIII. Addressing Syntactic Drift Over Time

Due to the limited size and temporal coverage of the available dataset, this section offers an exploratory analysis of the preliminary results that evaluate the temporal robustness of the proposed multi-modal architecture, which incorporates extended structured features.

TABLE XV. Temporal distribution of CVEs per ATT&CK tactic for the string-concatenated dataset.

Tactic	Train ($\leq$2021)	Validate (==2022)	Test ($\geq$2023)
Reconnaissance	85	4	52
Resource Development	83	1	33
Initial Access	483	72	18
Execution	1084	83	16
Persistence	1584	7	0
Privilege Escalation	1692	35	4
Defence Evasion	5346	8	0
Credential Access	515	17	2
Discovery	1956	3	0
Lateral Movement	614	5	1
Collection	558	17	1
Command and Control	380	1	1
Exfiltration	70	29	27
Impact	260	25	1

TABLE XVI. Temporal distribution of CVEs per ATT&CK tactic for the multi-modal dataset.

Tactic	Train ($\leq$2021)	Validate (==2022)	Test ($\geq$2023)
Reconnaissance	85	4	52
Resource Development	83	1	37
Initial Access	487	72	20
Execution	1086	83	22
Persistence	1580	8	0
Privilege Escalation	1694	36	6
Defence Evasion	5319	9	0
Credential Access	509	17	2
Discovery	1938	3	0
Lateral Movement	613	5	2
Collection	554	17	1
Command and Control	379	1	1
Exfiltration	70	29	29
Impact	255	25	1

It is important to note that the multi-modal dataset was constructed in early 2026, while the string-concatenated dataset was created in early 2025. In the meantime, approximately 12 additional CVEs have been analyzed by the National Vulnerability Database (NVD). This provides our multi-modal dataset with a slight increase in test data; however, given the relatively small amount of additional data, the results presented here remain comparable.

A. Performance Metrics

Table XVII summarizes the Micro, Macro, and weighted F1 scores for the temporally split analysis of both the string-concatenated and multi-modal models. These metrics are meaningful measures of effectiveness, especially considering the sparsity of the test data (post-2022). The multi-modal model consistently outperforms the string-concatenated model, showing significant improvements in both Micro and Weighted F1 scores. This indicates enhanced overall predictive performance and better generalization, even with limited data. These results support earlier findings in this study, suggesting that including structured vulnerability features provides additional, temporally stable signals that enhance robustness to syntactic drift. Given the small size of the post-2022 test split, the findings should be interpreted as exploratory. Nonetheless, they provide evidence in favour of the proposed multi-modal approach.

TABLE XVII. Temporal test-set performance comparison between string-concatenated and multi-modal fine-tuned models.

Model	Micro F1	Macro F1	Weighted F1
String Concatenated	0.143	0.080	0.179
Multi-Modal	**0.300**	**0.172**	**0.391**

B. Discussion

The temporal evaluation highlights the challenges of generalising across evolving vulnerability descriptions. However, we present preliminary evidence that the proposed multimodal architecture is more resilient to syntactic drift than a purely text-based baseline. As indicated in Table XV and Table XVI, the post-2022 test split is both sparse and highly imbalanced across tactics. This limits the absolute performance achievable by either model and underscores the need to use F1-based metrics rather than accuracy.

Despite these limitations, the multimodal model shows consistent improvements across all aggregated metrics (refer to Table XVII), particularly in weighted F1. This suggests that incorporating structured vulnerability metadata yields temporally stable signals that enhance natural language descriptions, even as phrasing evolves. The increase in macro F1 further indicates improved generalisation to less frequent tactics, aligning with previous observations that structured features are particularly beneficial for minority and semantically ambiguous classes.

Per-tactic analysis (see Table XVIII) supports this conclusion. Notable improvements for tactics such as *Resource Development*, *Initial Access*, and *Exfiltration* indicate that the multimodal model is better equipped to leverage auxiliary signals, such as exploitability and impact, when textual evidence alone is insufficient or inconsistent. The lack of meaningful performance for several tactics reflects extreme data sparsity in the temporal test split rather than a failure of the model, highlighting the challenge of evaluating temporal robustness in real-world CVE datasets.

Overall, these results align with the feature importance analysis presented earlier (see Tables XI, XII, XIII, and XIV), reinforcing the conclusion that the multimodal architecture does not replace textual understanding but instead enhances

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

it with structured, semantically grounded signals that remain informative over time. While these findings should be interpreted as exploratory, they provide compelling evidence that a multimodal representation offers a more robust foundation for long-term mapping of CVE-to-ATT&CK tactics.

TABLE XVIII. PER-TACTIC F1 SCORES ON THE POST-2022 TEST SPLIT FOR BOTH MODELS.

Tactic	String Concatenated	Multi-Modal
Resource Development	0.706	**0.914**
Initial Access	0.071	**0.267**
Execution	0.167	**0.225**
Privilege Escalation	0.182	**0.231**
Exfiltration	0.000	**0.766**

N.B. Tactics for which both models achieved an F1 score of 0.00 are omitted. In these cases, the post-2022 test split contained very few instances, resulting in unstable per-class metrics that are not statistically meaningful.

IX. LIMITATION

A significant limitation of this research is the dynamic nature of the MITRE ATT&CK framework and the rapid evolution of the cyber threat landscape. This may result in misalignment between the framework and the most current adversary tactics, techniques, and procedures (TTPs).

Despite the enhanced approach and the shift to a multi-modal architecture, the *Reconnaissance* tactic remains particularly challenging to predict. While the under-representation of this tactic in the dataset could explain the difficulties, this reasoning is insufficient. In contrast, the *Exfiltration* tactic, similarly represented (as shown in Table II), exhibits significantly greater predictability.

Another limitation is the exclusion of CVEs that lack extended fields. This exclusion introduces bias toward well-documented vulnerabilities and leaves out zero-day threats, which cannot be included until they are evaluated by the National Vulnerability Database (NVD) and assigned relevant attributes, such as Common Platform Enumeration (CPE). Although the multi-modal architecture enhances performance, it relies on the timely analysis of newly published CVEs in the NVD to integrate these auxiliary signals into the pipeline. To address this issue, future work will explore partial feature selection to broaden coverage until a fully extended dataset becomes available. Additionally, large language models (LLMs) can improve vulnerability descriptions by leveraging historical data, enabling the system to understand new vulnerabilities without retraining.

Lastly, while SHAP-inspired global feature importance analysis offers valuable insights into the model's aggregate behaviour, it does not fully capture instance-level decision dynamics or the causal relationships between features and predictions. Therefore, the results of interpretability analyses should be regarded as indicative rather than definitive explanations of the model's reasoning.

X. CONCLUSION AND FUTURE WORK

This research demonstrates that augmenting CVE descriptions with additional structured vulnerability metadata, including EPSS, CWE, CVSS, CPE, and CAPEC, significantly improves both the accuracy and robustness of automated mapping to MITRE ATT&CK tactics. Across unsupervised and supervised experimental settings, feature enrichment consistently reduced misclassifications and improved correct mappings, with the most pronounced gains observed when textual descriptions were sparse, ambiguous, or inconsistently written. From a practitioner's perspective, this finding has significant operational implications. Vulnerability reports of poor quality, which is a pervasive reality in production CVE feeds, should not limit the reliability of mapping when structured metadata is available to provide support.

In supervised experiments, the proposed SecRoBERTa-based approach outperformed existing state-of-the-art models, and further improvements were achieved through systematic hyperparameter optimisation. More importantly, the shift from a string-concatenated representation to a multi-modal architecture allowed structured metadata to serve as a numerically stable auxiliary signal rather than being weakly encoded through natural language. Feature importance analysis confirmed that, while textual information remains the dominant driver of predictions, structured features provide complementary, semantically grounded signals that are especially beneficial for minority tactics and those historically difficult to predict. In operational terms, this means more reliable tactic coverage across the ATT&CK framework, including adversarial behaviours that are consistently underrepresented in training datasets yet highly relevant to real-world threat landscapes.

The temporal evaluation further underscored the practical value of this architectural shift. Despite facing severe class imbalance and data sparsity in the post-2022 test split, the multi-modal model demonstrated improved generalization and resilience to syntactic drift in evolving vulnerability descriptions. These findings suggest that incorporating structured vulnerability metadata provides a more future-resilient foundation for automated CVE-to-ATT&CK mapping than text-only approaches. This resilience is especially important for Security Operations Centers (SOCs), where it is vital for mapping pipelines to remain reliable as the Common Vulnerabilities and Exposures (CVE) ecosystem evolves and descriptive conventions change. A model that can adapt smoothly to temporal distribution shifts, rather than failing dramatically, reduces the operational burden of periodic retraining and manual recalibration, supporting more sustainable long-term deployment.

The main practical contribution of this work is its direct alignment with threat-informed vulnerability prioritization. By accurately and consistently mapping CVEs to adversarial tactics, the proposed framework allows security teams to move beyond relying solely on severity scores for triage. Instead, it enables them to explicitly consider how each vulnerability fits into known attack chains. In resource-constrained environments, where remediation capacity is limited and adversarial threats are constant, this capability supports more strategically grounded risk assessments and leads to more defensible prioritization decisions. Integrating this framework into current

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

vulnerability management workflows could significantly reduce the time between vulnerability disclosure and informed defensive responses.

Future work will focus on extending this methodology in several directions. First, a dedicated CAPEC integration pipeline or API will be developed to enable richer and more consistent incorporation of attack pattern data. Second, building on the strong performance at the tactic level, this research will be extended to fine-grained ATT&CK technique classification using a constrained supervised approach, in which predictions are limited to techniques associated with the predicted parent tactic.

Beyond direct feature enrichment, future research will explore methods to identify and augment missing or incomplete CVE attributes. This includes predicting missing structured fields using machine-learning models trained on historical vulnerability data and leveraging Large Language Models to infer missing contextual information from newly disclosed CVEs. Such enhancements have the potential to improve not only CVE-to-ATT&CK mapping but also downstream applications such as exploitability prediction, automated vulnerability triage, and adversary behaviour modelling.

Overall, this work provides evidence that multi-modal representations offer a more accurate, interpretable, and temporally robust approach to CVE-to-ATT&CK mapping, establishing a foundation for future research at the intersection of vulnerability intelligence, adversary modelling, and applied machine learning. Furthermore, it offers a technically advanced and operationally relevant pathway for developing more effective automated cyber defense strategies.

Acknowledgments

This work was supported by the NICYBER2025 programme funded by Innovate UK.

References

[1] L. McMahon and O. Olukoya, "Threat-Based Vulnerability Management: Mapping CVEs to the MITRE ATT&CK Framework," in *The Nineteenth International Conference on Emerging Security Information, Systems and Technologies: SECURWARE 2025*, IARIA, 2025.

[2] The MITRE Corporation, "Metrics," Retrieved: July 2025, 2025. [Online]. Available: https://www.cve.org/about/Metrics.

[3] Wade Baker, "The Pithy P2P: 5 years of vulnerability remediation & exploitation research," Retrieved: July 2025, 2025. [Online]. Available: https://www.cyentia.com/pithy-p2p/.

[4] P. Garrity, "State of Exploitation - A Peek into the Last Decade of Vulnerability Exploitation," Retrieved: July 2025, 2024. [Online]. Available: https://vulncheck.com/blog/state-of-exploitation-a-decade.

[5] Cyentia Institute, "A Visual Exploration of Exploitation in the Wild: The Inaugural Study of EPSS Data and Performance," Retrieved: July 2025, 2024. [Online]. Available: https://www.cyentia.com/wp-content/uploads/2024/07/EPSS-Exploration-Of-Exploits.pdf.

[6] Google Cloud Security, "Mandiant M-Trends 2025 Report," Retrieved: September 2025, 2025. [Online]. Available: https://services.google.com/fh/files/misc/m-trends-2025-en.pdf.

[7] J. Baker, "2023 R&D Roadmap to Advance Threat-Informed Defense," Retrieved: July 2025, 2023. [Online]. Available: https://medium.com/mitre-engenuity/2023-r-d-roadmap-to-advance-threat-informed-defense-cf726d30e583.

[8] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," in *Technical report*, The MITRE Corporation, 2018.

[9] The MITRE Corporation., "Att&ck®," Retrieved: August 2025, 2025. [Online]. Available: https://attack.mitre.org/.

[10] J. Baker, "CVE + MITRE ATT&CK® to Understand Vulnerability Impact," Retrieved: July 2025, 2021. [Online]. Available: https://medium.com/mitre-engenuity/cve-mitre-att-ck-to-understand-vulnerability-impact-c40165111bf7.

[11] D.-Y. Kim, S.-S. Yoon, and I.-C. Euom, "V2TSA: Analysis of Vulnerability to Attack Techniques using a Semantic Approach," *IEEE Access*, 2024.

[12] O. Grigorescu, A. Nica, M. Dascalu, and R. Rughinis, "CVE2ATT&CK: BERT-Based Mapping of CVEs to MITRE ATT&CK Techniques," *Algorithms*, vol. 15, no. 9, p. 314, 2022.

[13] A. Kuppa, L. Aouad, and N.-A. Le-Khac, "Linking CVE's to MITRE ATT&CK Techniques," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–12.

[14] B. Abdeen, E. Al-Shaer, A. Singhal, L. Khan, and K. Hamlen, "SMET: Semantic Mapping of CVE to ATT&ck and its Application to Cybersecurity," in *IFIP Annual Conference on Data and Applications Security and Privacy*, Springer, 2023, pp. 243–260.

[15] I. Branescu, O. Grigorescu, and M. Dascalu, "Automated Mapping of Common Vulnerabilities and Exposures to MITRE ATT&CK Tactics," *Information*, vol. 15, no. 4, p. 214, 2024.

[16] B. Ampel, S. Samtani, S. Ullman, and H. Chen, "Linking Common Vulnerabilities and Exposures to the MITRE ATT&CK Framework: A Self-Distillation Approach," *arXiv preprint arXiv:2108.01696*, 2021.

[17] O. Mendsaikhan, H. Hasegawa, Y. Yamaguchi, and H. Shimada, "Automatic Mapping of Vulnerability Information to Adversary Techniques," in *The Fourteenth International Conference on Emerging Security Information, Systems and Technologies SECUREWARE2020*, 2020, pp. 53–59.

[18] Vulcan Cyber, "CVE to T&TS: Using CVE attributes for MITRE ATT&CK Mapping," Retrieved: July 2025, 2023. [Online]. Available: https://web.archive.org/web/20240429155732/https://l.vulcan.io/hubfs/Ebooks-and-White-Papers/Vulcan-Cyber-Mapping-CVEs-to-MITRE.pdf.

[19] C. Adam, M. F. Bulut, D. Sow, S. Ocepek, C. Bedell, and L. Ngweta, "Attack Techniques and Threat Identification for Vulnerabilities," *arXiv preprint arXiv:2206.11171*, 2022.

[20] Q. Li, W. Tang, X. Chen, and H. Ren, "VuldiffFinder: Discovering inconsistencies in unstructured vulnerability information," *Computers & Security*, p. 104 447, 2025.

[21] Y. Dong, W. Guo, Y. Chen, X. Xing, Y. Zhang, and G. Wang, "Towards the Detection of Inconsistencies in Public Security Vulnerability Reports," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 869–885.

[22] Y. Chen, A. E. Santosa, A. Sharma, and D. Lo, "Automated Identification of Libraries from Vulnerability Data," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice*, 2020, pp. 90–99.

[23] H. Guo, S. Chen, Z. Xing, X. Li, Y. Bai, and J. Sun, "Detecting and Augmenting Missing Key Aspects in Vulnerability Descriptions," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 31, no. 3, pp. 1–27, 2022. DOI: 10.1145/3498537.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[24] T. Chen et al., "VulLibGen: Generating Names of Vulnerability-Affected Packages via a Large Language Model," *CoRR*, 2023.

[25] Center for Threat-Informed Defense, "Mapping MITRE ATT&CK® to CVEs for Impact," Retrieved: July 2025, 2021. [Online]. Available: https://github.com/center-for-threat-informed-defense/attack_to_cve.

[26] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.

[27] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of cti sources," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, 2017, pp. 103–115.

[28] V. Legoy, M. Caselli, C. Seifert, and A. Peter, "Automated Retrieval of ATT&CK Tactics and Techniques for Cyber Threat Reports," *arXiv preprint arXiv:2004.14322*, 2020.

[29] J. Fairbanks, A. Orbe, C. Patterson, J. Layne, E. Serra, and M. Scheepers, "Identifying ATT&CK Tactics in Android Malware Control Flow Graph Through Graph Representation Learning and Interpretability," in *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, 2021, pp. 5602–5608.

[30] S. Vasan et al., "DEEPCAPA: Identifying Malicious Capabilities in Windows Malware," in *2024 Annual Computer Security Applications Conference (ACSAC)*, IEEE, 2024, pp. 826–842.

[31] Y. Sharma, S. Birnbach, and I. Martinovic, "RADAR: A TTP-based Extensible, Explainable, and Effective System for Network Traffic Analysis and Malware Detection," in *Proceedings of the 2023 European Interdisciplinary Cybersecurity Conference*, 2023, pp. 159–166.

[32] M. Lv et al., "Trec: Apt tactic / technique recognition via few-shot provenance subgraph learning," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 139–152.

[33] P. Bhosale, W. Kastner, and T. Sauter, "Mapping ICS Vulnerabilities: Prioritization and Risk Propagation Analysis with MITRE ATT&CK Framework and Bayesian Belief Networks," in *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2024, pp. 1–8.

[34] The MITRE Corporation (MITRE), "Common Weakness Enumeration," Retrieved: September 2025, 2025. [Online]. Available: https://cwe.mitre.org/.

[35] Forum of Incident Response and Security Teams, "Common Vulnerability Scoring System," Retrieved: July 2025, 2025. [Online]. Available: https://www.first.org/cvss/.

[36] NIST, "Official Common Platform Enumeration (CPE) Dictionary," Retrieved: July 2025, 2025. [Online]. Available: https://nvd.nist.gov/products/cpe.

[37] J. Jacobs, S. Romanosky, B. Edwards, I. Adjerid, and M. Roytman, "Exploit Prediction Scoring System (EPSS)," *Digital Threats: Research and Practice*, vol. 2, no. 3, pp. 1–17, 2021.

[38] The MITRE Corporation, "Common Attack Pattern Enumeration and Classification," Retrieved: July 2025, 2023. [Online]. Available: https://capec.mitre.org/.

[39] NIST, "Vulnerabilities: CVE API," Retrieved: July 2025, 2025. [Online]. Available: https://nvd.nist.gov/developers/vulnerabilities.

[40] Forum of Incident Response and Security Teams, "EPSS API," Retrieved: July 2025, 2025. [Online]. Available: https://www.first.org/epss/api.

[41] Y. Lakhdhar and S. Rekhis, "Machine Learning Based Approach for the Automated Mapping of Discovered Vulnerabilities to Adversial Tactics," in *2021 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2021, pp. 309–317.

[42] NIST, "Common Vulnerability Scoring System Calculator - CVSS Version 2.0," Retrieved: July 2025, 2025. [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator.

[43] NIST, "Common Vulnerability Scoring System Calculator - CVSS Version 3.0, CVSS Version 3.1," Retrieved: July 2025, 2025. [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.

[44] Kun jackaduma, "SecRoBERTa," Retrieved: July 2025, 2023. [Online]. Available: https://huggingface.co/jackaduma/SecRoBERTa.

[45] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint arXiv:1907.11692*, 2019.

[46] F. Castaño, A. Gil-Lerchundi, R. Orduna-Urrutia, E. F. Fernandez, and R. Alaiz-Rodríguez, "WAVE-27K: Bringing together CTI sources to enhance threat intelligence models," in *Proceedings of the First International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security*, 2024, pp. 119–126.

[47] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online: Association for Computational Linguistics, 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6.

[48] T. Mimoto, Y. Gempei, K. Kita, T. Isohara, S. Kiyomoto, and T. Tanaka, "Linkage between cve and att&ck with public information.," in *SECRYPT*, 2024, pp. 655–660.

[49] J. Evans, "MITRE Key Details Phrasing," Retrieved: September 2025, 2020. [Online]. Available: https://cveproject.github.io/docs/content/key-details-phrasing.pdf.

[50] C. Madden, "Vulnerability Description Quality Checks and Data Analysis," Retrieved: September 2025, 2025. [Online]. Available: https://github.com/CyberSecAI/VulnerabilityDescriptionQualityChecker.

[51] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[52] Z. Zhang, H. Al Hamadi, E. Damiani, C. Y. Yeun, and F. Taher, "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research," *IEEE Access*, vol. 10, pp. 93 104–93 139, 2022.

[53] M. Aota, H. Kanehara, M. Kubo, N. Murata, B. Sun, and T. Takahashi, "Automation of Vulnerability Classification from its Description using Machine Learning," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2020, pp. 1–7. DOI: 10.1109/ISCC50000.2020.9219568.

[54] Z. Han, X. Li, Z. Xing, H. Liu, and Z. Feng, "Learning to Predict Severity of Software Vulnerability Using Only Vulnerability Description," in *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, IEEE, 2017, pp. 125–136. DOI: 10.1109/ICSME.2017.52.

[55] E. Iannone, G. Sellitto, E. Iaccarino, F. Ferrucci, A. De Lucia, and F. Palomba, "Early and Realistic Exploitability Prediction of Just-Disclosed Software Vulnerabilities: How Reliable Can It Be?" *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 6, pp. 1–41, 2024. DOI: 10.1145/3654443.

[56] J. Jacobs, S. Romanosky, O. Suciu, B. Edwards, and A. Sarabi, "Enhancing Vulnerability Prioritization: Data-Driven Exploit Predictions with Community-Driven Insights," in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, IEEE, 2023, pp. 194–206.

[57] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

MOTRA: Data Generation for HIDS & NIDS in OT-focused Environments

Peter Heller, Sebastian Kraust, and Jürgen Mottok
Laboratory for Safe and Secure Systems (LaS³)
OTH Regensburg
93053 Regensburg, Germany
e-mail: {peter2.heller | sebastian.kraust | juergen.mottok}@oth-regensburg.de

***Abstract*—As critical infrastructure faces increasingly sophisticated attacks from different domains, the research community requires testbeds that can adapt to changing threat landscapes to generate high-quality, reproducible datasets. Existing testbeds are often monolithic, rigid, or dependent on legacy protocols, hindering the development of modern intrusion detection systems. Our work introduces MOTRA (MOdular Testbed for Researching Attacks), a fully open-source containerized security testbed to simulate attacks on industrial protocols. By employing container-based orchestration and a custom, time-synchronized automation framework, MOTRA enables the rapid configuration of complex industrial control system environments. We introduce a capture framework based on a client-server architecture, capable of orchestrating both benign operational interactions and complex attack vectors to facilitate the generation of pristine host and network captures, including command and control and man-in-the-middle setups. We present a preliminary dataset that contains network captures in pcap format and host captures in JSON format using perf data. MOTRA provides a proof of concept to create reproducible configurations for dataset generation covering multi-stage, evolving cyber-physical threats.**

***Keywords*-OT-Testbed; OPC UA; cybersecurity; data generation; industrial security.**

I. Introduction

The increasing complexity and interconnectivity of Industrial Control Systems (ICS) require extensive security assessments. This is no trivial task considering the rapidly evolving attack surface, the widespread use of devices and protocols with limited security features, and the introduction of new security architectures that require complex dependencies. Penetration tests offer a way to actively assess a system's security capabilities but are impractical in live industrial environments, due to the risk of damaging equipment and endangering human life. Testbeds address this issue by providing a safe environment for evaluating security mechanisms, analyzing cyber-physical threats, and developing defense strategies. However, existing testbeds are often static in nature, tightly coupled to specific architectures, or require extensive effort to modify, making them unsuitable for adapting them to the specific needs of a given scenario or emerging threats. Most notably, this rigidity limits their usefulness for assessing threats to one's own system, which might be vastly different from a prefabricated testbed.

To address this issue, we introduce a novel Industrial Control Systems (ICS) testbed prototype focusing on OPC UA [2] as the core protocol. OPC UA is an open, scalable standard built on a layered architecture that supports different communication patterns, protocols, and security configurations. Our testbed design was created to allow for high modularity and flexibility, enabling rapid restructuring and reconfiguration to accommodate various attack vectors and system configurations. The basic idea is to provide a set of building blocks in the form of containerized applications, from which a wide variety of testbed compositions can be created with minimal effort. By making the testbed components freely available, we hope to facilitate the sharing of domain knowledge, attack configurations, and custom testbed extensions within the research community. To enhance ongoing research efforts for Host-based and Network-based Intrusion Detection Systems (HIDS, NIDS), we introduce MOTRA Capture, a measurement and automation framework, as an extension of our current testbed prototype. Using predefined capture configurations, our tool can provide repeatable measurements and interactions to create datasets for intrusion detection research. Our contributions can be summarized as follows.

- **C1** We introduce the MOTRA testbed and supplementary applications.
- **C2** We introduce our data generation workflow based on the MOTRA Capture framework to allow for unattended testbed execution and data acquisition across remote clients to generate HIDS and NIDS data.
- **C3** We present a preliminary dataset containing 155 samples for measuring baseline activity and different attack configurations (MITM, DoS, injection, reconnaissance, modification, credential leak, and command and control) inside the current testbed prototype.

The remainder of this work is organized as follows. Section II covers testbed literature and a detailed comparison of available design criteria. Section III introduces historical context on previous malware campaigns targeting critical infrastructure, recent developments in OT malware, and the current state of existing datasets. The design choices for the MOTRA testbed are discussed in Section IV, which also addresses the integration of OPC UA, technical challenges and requirements that arose during the development process. Section V covers the current state of our testbed prototype. Section VI covers a hands-on example of how to set up, configure, and run testbed components for real-world use. Section VII introduces the capture framework used to generate host and network datasets within our testbed using different types of attack samples that we publish as part of our framework. Last, Section VIII covers the currently implemented types of attacks and configurations that are provided by our capture framework.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

II. Related Work

We focus on the integration or development of specific data acquisition systems in the context of HIDS and NIDS data generation and testbed orchestration for cybersecurity-related use cases. In this context, we primarily review testbed implementations for containerized solutions, virtualization, or automation, as these are core concepts that we addressed as part of our previous work on the MOTRA testbed [1]. KYPO4INDUSTRY [3], minicps [4], LabShock [5], and Emuflex [6] are testbeds that share similarities in the use of containerization with our current approach. Testbeds built on virtualization with additional requirements for automation or orchestration are GRFICS [7], RICS-el [8], VTET [9], Maynard [10], and DHALSIM [11]. In the following section, we will explore the available testbed literature in detail, highlight differences, and discuss relevant design choices.

A. Virtual OT Testbeds with Focus on Container Technology, Automation, and Security

KYPO4INDUSTRY [3] is an educational testbed that covers a number of virtual PLCs and different industrial protocols. The testbed is used as part of a university course program in a hybrid configuration using gamification techniques to recreate different types of historical cyberattacks from public literature. As part of the syllabus, students can present and develop different types of real-world attacks with differing levels of complexity. The testbed's core architecture uses virtualization for deployment of devices, workstations, Programmable Logic Controler (PLC) and backend services. The orchestration of the testbed is done separately by a virtual appliance to monitor the entire system and the available devices. Containers are used to deploy different virtual PLC devices, adopt emerging protocols (e.g., REST or MQTT), and introduce missing technologies. Some sensors and actuators are implemented in hardware as part of the main control panel to test cyber-physical aspects. While the general use of virtualization technologies and containerization is closely related to our use cases, there is no focus on generating new data sets for intrusion detection applications or anomaly detection. Additionally, reproducibility is irrelevant as the configuration and design details are beyond the scope of the publication. The primary focus is on the university course curriculum while providing valuable insight on design decisions and requirements from the development stage of the testbed.

minicps [4] is a testbed that enables the creation, sharing, and building of complex network emulations and device simulations using a Python framework. An advantage of this type of testbed is the ability to create complex configurations at the device and network architecture level. This allows for dynamic testing and sharing of different testbed architectures among researchers. The core framework uses mininet [12] as a backend and relies on a Software Defined Networking (SDN) approach to emulate networks and device interfaces. The core architecture currently supports Ethernet/IT and Modbus/TCP but should work with other IP-based protocols. In addition, minicps adds a custom API to allow third-party physical simulations to extend the testbed for better domain fidelity. Their core architecture focuses on insecure-by-design protocols, like Ethernet/IP and Modbus. These protocols do not support state-of-the-art security configurations and offer less value in the testing of modern communication architectures or security methodologies. Implementing complex attacks using different levels of security configurations is not possible with this testbed design. Another important issue is the current state of the used emulation platform. Judging by the latest activity on the project homepage and the state of issues and releases on GitHub, it seems that mininet [12] is not being actively maintained anymore.

LabShock [5] is one of the first freely available (partly open source) testbeds for hands-on security testing. The entire toolkit comprises various modules that can be used for testing, as well as a centrally managed portal where OT-specific tests can be completed at several difficulty levels. It utilizes various open-source containers and a proprietary backend with its proprietary license. The aim of the testbed is hands-on learning in private and corporate environments. Its advantage is the variety of available tools and integrations, as Demilitarized Zone (DMZ), enterprise connectivity, Security Information and Event Management (SIEM), and Intrusion Detection Systems (IDS) can be tested in a preconfigured environment. While the general modules and interfaces are all open source, the specific backends of the framework are proprietary. This has implications for reproducibility, since there is little information available on how these systems behave in detail and how the overall project develops over time. Although the level of orchestration and guided tests are very valuable as an educational resource, the current design and proprietary modules of the testbed do not allow for accurate recording of data sets for IDS research due to the questionable quality of the data. However, the testbed could be used for proof-of-concept attacks on simulated systems.

Emuflex [6] is a small fully virtualized framework that allows Small-Medium-sized Enterprises (SME) to perform hands-on security training, integration tests, and general security experimentation. The focus is on modularity and usability to provide SMEs with a valuable training resource. The core architecture is built on OPC UA and Modbus/TCP and offers a number of basic IIoT blocks: Human Machine Interface (HMI), PLCs (OpenPLC), networked devices, and a data historian for event logging and analysis. The entire setup is enclosed inside a single deployment container to load different system architectures, test components, and attacks. The core stack supports a custom industrial process based on a pressurized water tank. For better insight, the simulation module is decoupled from the process visualizer, allowing an operator to check on simulation changes during an active attack. Notably, the setup offers Certificate Authority (CA) and certificate management functionality. However, since this testbed is designed only for educational hands-on training purposes, very few design choices have been made to support reproducible attacks for dataset creation.

DHALSIM [11] is a high-fidelity simulation framework that

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

couples network emulation of minicps and mininet with the physical simulation components of the WNTR framework. Their digital twin approach builds a reproducible testbed that allows them to simulate attacks with physical process accuracy. The main focus of the work is to extend digital twin functionality to a cybersecurity experimentation platform. Due to the extension of minicps, the framework can simulate a number of networked industrial devices and capture network traffic at the interface level of each simulated node. The authors test different real-world attacks and run Man-In-The-Middle (MITM) payloads to perform concealment attacks on their simulated targets. While the framework itself offers many research opportunities for dataset creation, future development is currently impacted by the lack of maintenance of mininet.

GRFICS [7] is a virtualized testbed that has been designed as an educational resource for hands-on training. The core architecture provides a custom Tennessee Eastman Process (TEP) simulation that integrates with a virtualized PLC and HMI. At the protocol level, this testbed supports Modbus/TCP. In addition, the full-scale system is 3D-rendered using a custom graphical engine to allow an operator to verify how the simulated system behaves during an attack. As a result, physical out-of-bounds operations can be visualized as part of the experimental setup. Notably, this testbed uses a number of different attacks, ranging from OT-specific protocol injections to IT-specific attacks such as brute force, buffer overflow, or credential leaks. From the initial release, the testbed has been maintained and updated to include monitoring tools, a firewall configuration with integrated DMZ, and an automated adversary emulation using MITRE Caldera. The latest release runs on Docker and Docker Compose and virtualizes all test instances locally. The core application currently only works with Modbus/TCP, limiting testing capabilities to insecure protocol configurations.

RICS-el [8] is a full-scale virtualized testbed that covers enterprise devices and services, a simulated internet connection, different layers of DMZ, and an OT segment made up of different process zones. This level of design has been achieved by including expert knowledge from both industry and research groups to build a high-fidelity system to conduct cybersecurity research for mitigation strategies and dataset creation. The OT segment covers a power grid simulator and redundant SCADA servers that are used to train operators and can provide different operational scenarios. In particular, the DMZ between enterprise IT and SCADA systems is designed to exchange historical station data and generation schedules in a secure manner to simulate real traffic. Different nodes inside the testbed are developed as worker bots to create additional live traffic to be used for high-fidelity datasets. Although the design criteria and practical experiences are very valuable, the implementation scope is currently limited to the IEC 60870-5-104 protocol.

VTET [9] is another virtualized testbed built using the Tennessee Eastman process in a virtualized or hybrid configuration to perform cybersecurity experiments in a safe manner. Two examples are shown using either a virtualized PLC or a real Siemens PLC for their experiments. The core architecture uses a MATLAB toolbox to provide the simulation backend for the chemical process and two PLCs to perform control operations to keep the simulated backend within safe boundaries. For communication between devices and the simulation toolbox, OPC UA, Modbus, and Siemens S7Comm are used to exchange data and control messages. Although the setup generally offers high fidelity due to the use of physical simulation and real hardware, we were unable to find detailed configurations or freely available tools to rebuild the testbed. Additional material would enhance the reproducibility and credibility of this setup. Another important aspect is the provided attacks used to verify the general approach. While the different physical aspects of the attacks are very useful for a cyber-physical experiment, nearly all the attacks build on the same design weakness of similar protocols. Finally, the authors mention the impractical use of the provided attacks, and therefore different attack patterns would be a valuable addition to the testbed.

The authors of Maynard SCADA [10] present an open-source experimental framework that aids researchers in deploying custom cybersecurity Supervisory Control and Data Aquisition (SCADA) testbeds. The reference architecture for the testbed covers different virtual and physical devices typically found inside a SCADA environment: PLCs, Remote Terminal Unit (RTU), HMI, Historian, sensor, and actor. At the protocol level, the testbed covers OPC UA, IEC61850, Modbus/TCP, and 60870-5-104 (IEC104). What makes this work stand out is the detailed description of how the orchestration layer was built and designed, as well as the comparison of testbed features against state-of-the-art requirements found in available literature. In general, the presented setup consisting of nine hosts is used to create a dataset that contains five different attacks on ICS devices. The testbed as well as the orchestration service are available as open-source software, allowing other researchers to rebuild and extend the available tools as needed. At the time of writing, the latest available release of the testbed and orchestration layer was published about eight years ago.

B. Detailed Comparison of Existing Testbed Literature

To provide a detailed comparison, we have outlined the key aspects of each study in the tables below. Table I first presents the most important general parameters, such as scope, coverage according to the Purdue Enterprise Reference Architecture (PERA) model, target domain, and IT-OT domain crossover. For feature selection, we relied on the work of Conti et al. [13] for general requirements, design choices, and best practices, and Murillo et al. [11] for representation characteristics for ICS, network, and physical aspects. It is worth noting that the majority of the virtual OT testbeds examined are freely accessible and available under open-source licenses.

Based on the general information we collected in Table I, we can make assumptions about the scope and architecture of available testbeds related to available PERA layer coverage. Covering multiple layers is particularly relevant today, as

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. OVERVIEW OF OT TESTBEDS AND CHARACTERISTICS. NOTE: DEMILITARIZED ZONE CROSSOVER INCLUDED (DMZ), OT SECURITY (OT), CYBERSECURITY (CS), CYBER-PHYSICAL (CP), EDUCATIONAL (ED), SIMULATED (S), HYBRID (H), COVERED (●), FULLY OPEN SOURCE (⬤), PARTLY OPEN SOURCE OR MISSING INFORMATION (◑), OPEN DESCRIPTION (○), NOT COVERED/INFORMATION NOT AVAILABLE (-).

Ref	Testbed	Year	License	DMZ	Scope				Type		PERA Level						Shareability
					OT	CS	CP	ED	S	H	0	1	2	3	4	5	
[4]	minicps	2015	MIT	-	-	-	●	-	●	-	●	●	●	●	-	-	⬤
[7]	GRFICS	2018	GPL3	-	●	-	-	●	●	-	●	●	●	-	-	-	⬤
[9]	VTET	2018	-	-	●	●	●	-	-	●	●	●	-	-	-	-	○
[8]	RICS-el	2019	-	●	●	●	●	-	●	-	●	●	●	●	●	●	○
[3]	KYPO4INDUSTRY	2020	MIT	-	●	-	-	●	●	●	●	●	●	-	-	●	◑
[11]	DHALSIM	2021	MIT	-	●	-	●	-	●	-	●	●	●	-	-	-	⬤
[6]	EmuFlex	2024	Apache 2.0	-	●	●	-	-	●	-	-	●	●	●	-	-	⬤
[10]	Maynard SCADA	2024	GPL3	-	●	●	-	-	-	●	-	●	●	●	-	-	⬤
[1]	MOTRA	2025	GPL3	●	●	●	-	-	●	●	-	●	●	●	-	●	⬤
[5]	LabShock	2026	proprietary	●	●	●	-	●	●	-	●	●	●	●	●	-	◑

modern industrial systems are becoming increasingly interconnected. Complex multi-stage attacks can only be realistically simulated when testbeds incorporate all levels of the PERA architecture. LabShock [5] and, in particular, RICS-el [8] integrate a wide range of components and cover nearly all PERA layers. This enables realistic modeling of domain crossover, representing the transition between IT and OT systems. According to current literature, such transitions are essential for researching security strategies and defense-in-depth solutions for IT-OT convergence, which are in turn required for generating high-quality realistic datasets. This becomes increasingly important considering the target scope outside the cybersecurity domain, as the current threat landscape changes and testbeds have a compelling need to improve. As Table I shows, this critical transition is currently only addressed by three of our chosen testbeds (MOTRA, LabShock & RICS-el), with the work of Almgren et al. [8] providing additional details on the subject.

As shown in Table I, most of the testbeds presented are covered by an open-source license and offer accessible software components. There are currently only two testbeds that are categorized as open description in our case: VTET and RICS-el. Although the previous work for VTET is available publicly to a certain extent, we could not find any online documentation for the detailed configuration and software components at the time of publication. RICS-el is part of the Resilient Information and Control Systems (RICS) research centre and, as far as we could tell, only publishes their data and security research. LabShock components are designed to be used in a commercial training setting. Therefore, the testbed uses a proprietary backend and component executables. They do offer a free tier and open-source configurations for individual containers, but the internal system and architecture are not available to the public. KYPO4INDUSTRY is different in this regard, as the educational environment and the lab configuration are documented comprehensively. However, the only downside to the current setting is the missing attack designs from the provided course program, which could be adapted to share more complex examples of real-world attacks.

For a more in-depth comparison, we turn to the more detailed architectural descriptions and representations collected in Table II. The work of Conti et al. [13] provides detailed best practices and general comparison criteria for testbeds. Murillo et al. [11] supplements this with a comparison in which existing literature based on digital twins is broken down into three relevant dimensions: network, ICS, and physics. These dimensions offer different levels of abstraction (i.e., nonexistent, simple model, simulator, emulator, real components) and can be related to the fidelity of a testbed. In theory, the higher the degree of fulfillment of the various dimensions, the better suited the environment is for generating high-quality training data for Intrusion Detection Systems (IDS) or anomaly detection.

A general comparison of the table provides two general categories of testbeds. On the one hand, there are educational testbeds (e. g., GRFICS [7], KYPO4INDUSTRY [3], LabShock [5]), which are designed for training and education purposes. The key differences here are the high level of abstraction of the industrial processes and the lack of interactions within the testbed, which makes them unsuitable for capturing high-quality training data for intrusion detection systems (IDS). On the other hand, there are testbeds specifically designed for data generation in the field of intrusion or anomaly detection, such as hybrid systems. They are designed using specific hardware and, through approaches such as emulation of the physical process and the network, are particularly well-suited for experiments in a cyber-physical context [1][3][9][10].

Differences can also be found in more complex testbeds, such as RICS-el [8], where secondary conversations and operator interactions are modeled within the network to generate a system as close to reality as possible for data collection. DHALSIM [11] uses multiple segmented zones and can simulate interactions at the network level using the hydraulic simulator. The MOTRA [1] testbed attempts to account for interactions within the testbed using the Capture Framework

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE II. DETAILED ARCHITECTURAL AND TECHNICAL REPRESENTATION OF OT TESTBEDS

Testbed	Physical Representation	Network Representation	ICS Representation	Orchestration	Protocols
[1] MOTRA	Python Script	Containerized Network	OPC UA Models, Python Scripts	Container Orchestration	OPC UA
[7] GRFICS	Custom TEP simulation using UNITY	Virtual Network	OpenPLC	no	Modbus
[10] Maynard SCADA	None	Virtual Network	OpenMUC for IEC 104, OPC UA	Custom Tools	IEC104, OPC UA
[9] VTET	TEP Simulink Toolbox	Physical Network	PLCSim, NetToPLCSim	no	S7Comm, Modbus, OPC UA
[8] RICS-el	400kV Power Grid Simulator	SDN, Virtual Network	Simulated RTUs for process WAN	FOI CRATE cyber range orchestrator	FTP, IEC 60870-5-104, NTP, Mail, Generic office traffic
[3] KYPO4INDUSTRY	Physical IO, COTS PLC devices	Containerized Network, Physical Network, SDN	OpenPLC, OpenStack	Virtual Appliance	–
[5] LabShock	Proprietary Simulation	Containerized Network	Modified OpenPLC	Container Orchestration	Modbus RTU/TCP, Ethernet/IP, BACnet IP, OPC UA, WebAPI, MQTT, S7
[4] minicps	Simulation API	Emulation (mininet)	Python Scripts	mininet	Ethernet/IP, Modbus/TCP
[6] EmuFlex	Python Script	Containerized Network	OpenPLC	Container Orchestration	OPC UA, Modbus
[11] DHALSIM	WNTR Framework	Emulation (mininet)	Python Script	mininet	TCP IP, Ethernet/IP, (Modbus)

for host-side and network-side data acquisition. In our opinion, interactions have generally been considered only in a few studies to date and would make an important contribution to high-quality datasets.

The available testbeds show a wide range of approaches regarding the characteristics of the physical process. For virtualized setups that closely mimic real-world physics, the trend is toward using high-quality simulation solutions. RICS-el [8], for example, uses a distribution network simulator that is intended for personnel training, while data generators such as DHALSIM [11] or minicps [4] connect to external physics simulators via API. Approaches in cybersecurity-oriented work (MOTRA [1], LabShock [5], EmuFlex [6]) tend to use basic backends for simulation, but, in return, allow for more configuration options within the testbed. In the case of the MOTRA [1] testbed, the Component Library allows various software configurations with known vulnerabilities to be used to simulate complex attacks within the testbed. In general, testbeds focusing primarily on security or education can be better adapted to specific applications, as these systems are usually less complex and extensive.

Despite these promising developments, there are limitations and challenges to the currently available solutions. A high level of fidelity often conflicts with flexibility and operating costs (technical debt). While a system like RICS-el [8] represents all PERA levels and uses high-quality physical simulations, it requires complex infrastructure for orchestration and operation. This leads to significant constraints when the system needs to be redesigned or expanded with new components. Particularly with regard to maintainability and future-proofing, the effort and costs associated with high-fidelity factors can become problematic in the future. As we can see with the current state of mininet [12], where updates and releases are stale, affecting downstream testbeds that depend on the emulation platform for future research.

Following an overview of current testbed literature and design representations, we shift our focus to the evolution of the threat landscape and the state of existing datasets in research. In order to establish a foundation for the subsequent sections, we introduce historical incidents and existing datasets to support the discussion of requirements and our own design decisions.

III. CURRENT STATE OF ATTACKS AND MALWARE CAMPAIGNS

The threat landscape for industrial systems continues to evolve. To understand how research and testbed environments need to be transformed, we discuss prior incidents, the state of known datasets, and recent academic work on malware in the OT domain.

A. Comparison of Real-World Incidents to Academic Work

Previous research has repeatedly examined large-scale attacks and malware campaigns in order to identify trends and observe changes in adversarial behavior. The authors of [14] have compiled a detailed overview of successful

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

and well-known cyberattacks on critical infrastructure and Cyber-Physical Systems (CPS). In addition, they broke down each incident according to impact on the PERA model for each attack or observed threat. This breakdown shows that almost all of the recorded attacks not only targeted individual system components but also intended to manipulate substantial portions of the downstream infrastructure and operations. In addition, we have included the work of [15] as a comparison to better understand the behavior of the currently well-known campaigns targeting critical infrastructure and CPS domains. As pointed out by [14], there have been notable trends in recent years. The complexity of malware and payloads is increasing. Instead of using individually designed toolkits for selective attacks, there is a tendency to use tools that are equipped with several individual payloads and are capable of dynamically reloading additional payloads on the fly. This makes sense as scale and heterogeneity in OT-specific environments increase with the addition of modern protocols and security architectures. In addition, certain toolkits have been designed to overcome the air-gapped portion of process networks and work in isolation in the segmented parts of the infrastructure.

TABLE III. SUMMARY OF MAJOR INCIDENTS AND MALWARE CAMPAIGNS FROM 2010 TO 2022 [14][15]. EACH INCIDENT HAS BEEN BROKEN DOWN INTO DOCUMENTED IMPACT AT THE PERA LEVEL (5-0) AND INITIAL ACCESS BASED ON THE FINDINGS OF [14]. INDICATIONS: KNOWN IMPACT ▼, INITIAL INFECTION ▲, SUSPECTED ENTRIES ▽, △.

	Year	5	4	3	2	1	0
Stuxnet	2010	▼▲	▼△		▼	▼	▼
DUQU	2011		▼▲				
SHAMOON	2011		▼▲				
HAVEX	2013	▼▲	▼	▼			
Black Energy (BE3)	2015	▲	▼	▼	▼		
INDUSTROYER	2016	▲	▼	▼	▼	▼	▼
KEMURI Water Company	2016	▼	▼		▼	▼	
TRITON	2017	△	△		▼	▼	
NotPetya	2017	▼▲	▼				
WannaCry	2017	△	▼△				
VPNFilter	2018		▽△	▽△			
Colonial Pipeline	2021	▼	▽△				
PIPEDREAM	2022		▽	▽	▽	▽	

We argue that the findings of Makrakis et al. [14] and Conti et al. [13], based on analyses of previous incidents and surveys of existing datasets and attack types, reveal a significant imbalance. Table III presents a summary of previous work on major incidents and highlights the impact and initial exposure of documented attacks. These evaluations rely on currently available forensic data and, in some cases, less accurate estimates. In certain instances, reports cannot provide definitive data on initial access and impact due to system damage or wiper malware. Therefore, we have noted instances where the data relies on such reconstructions or assumptions. Table IV shows a simplified version of the dataset summary from [13]. This table highlights the different attack classes that were used to build datasets for network-based intrusion detection systems. Conti et al. [13] cover a collection of 23 datasets that have been published in addition to existing testbed literature. With the primary focus on dataset creation for research on IDS and anomaly detection, Table IV shows a trend towards the use of Man-In-The-Middle (MITM) and injection-type attacks with less use of other potential scenarios. Given the nature of these attacks and the available attack types from Table IV, we argue that most covered attack scenarios rely on insecure protocol configurations, dated communication technologies, and cyber-physical characteristics. These types of attacks and configurations can be used to assess adversary behavior and impact close to the physical process or the process simulation (PERA levels 0, 1). If we compare this with the findings of recent incidents in Table III, we only get very limited coverage compared to existing real-world attacks, as shown before.

TABLE IV. SUMMARY OF AVAILABLE DATASETS BY CONTI ET AL. [13]. ATTACKS ARE GROUPED INTO THE FOLLOWING ENTRIES: RECONNAISSANCE (RC), REPLAY (RP), MAN-IN-THE-MIDDLE (M), INJECTION (I), DENIAL-OF-SERVICE (D), OTHERS (O).

Dataset Name	RC	RP	M	I	D	O
D5						
QUT_DNP3	●	●	●	●		●
QUT_S7Comm			●			
4SICS						
CyberCity Dataset	●		●	●	●	●
D2: Gas Pipeline				●		
D3b: Water S.T.	●			●	●	
D4: New Gas P.			●	●		●
Electra Modbus	●	●		●		
Electra S7Comm	●	●		●		
HVAC_Traces						
Lemay Covert						●
Lemay SCADA	●			●		●
Modbus SCADA N1			●		●	
S4x15 ICS						
WUSTL-IIOT-2018						●
D1: Power System				●		●
EPIC DATASET						
QUT_S7 (Myers)			●	●		
SWaT Dataset				●		●
BATADAL		●	●			●
HAI Dataset		●	●			
WADI Dataset			●			
Sum	**6**	**5**	**9**	**11**	**3**	**9**

A similar mismatch can be observed for emerging publications that involve attacks and exploitation originating from low-level ICS components. Existing works on PLC and control process security cover different types of attacks to disrupt control or safety functions [16]. When comparing recent publications on OT-specific attacks, we can observe tendencies targeting surrounding operational infrastructure. Senthivel et al. [17] present so-called denial of engineering operations attacks. Their proof of concept modifies an existing PLC program to infect a device with custom logic to allow an attacker to hide code changes and modifications from an operator. If an operator tries to access a PLC device, the custom malware can return either an unmodified program binary to the operator, effectively hiding any changes on

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

the PLC side, or send invalid data to crash the engineering software, disabling monitoring and debugging functionality for the operator. Similarly, Kalle et al. [18] present a proof of concept on how to inject a virtual PLC alongside a normal PLC program to conceal changes to existing PLC configurations or executed programs. This allows custom code to be executed on the infected PLC while maintaining a "normal" execution trace that will be used to interact with engineering or monitoring software. In this way, the virtual PLC can be used to interact with the operator while executing a different configuration on the real hardware. Other works cover the use of Command and Control (C2) functionality based on PLC devices. Klick et al. [19] show how control software can be utilized to run and hide a Socket Secure (SOCKS) proxy to pivot networks connected to a PLC device, effectively giving an attacker full access to an isolated production network. As shown before, the threat landscape extends beyond the cyber-physical domain and involves aspects that are currently not sufficiently covered by datasets.

B. Dataset Imbalances Compared to Current Threat Landscape

From Table III we can infer that attacks emerging from the enterprise domain primarily target downstream infrastructure based on previous research. We see a tendency for attacks to not focus on a single target but rather to prepare for more complex operations with the goals of lateral movement, execution, or command and control capabilities. In addition, certain attacks are actively trying to prevent forensic and recovery operations [14]. We do not encounter such interactions in the datasets currently presented. A significant portion of presented testbeds share a dependency on insecure protocols such as Modbus or basic Ethernet/IP. Especially in virtualized testbeds, it would be beneficial to use modern protocols or secured configurations to test mitigation strategies or simulate sophisticated attacks in addition to conventional technologies. We argue that it would be advisable to develop tools or benchmarks that can be verified in more comprehensive testbeds to create multidimensional datasets based on complex attack patterns and adversary interactions. Arguably, CPS injection, DoS, and MITM attacks are particularly relevant, as they are expected to have the greatest impact at the process level. In our opinion, the datasets are too biased in this respect if only the cyber-physical domain is considered long term. In the field of taxonomies, we have already observed work dealing with the distinction between cyber-to-cyber and cyber-to-physical attacks to create classifications based on cross-domain interaction. The work of [20] attempts to separate the available dimensions to compare threats in a way that cyber-to-physical, cyber-to-cyber, physical-to-physical, and physical-to-cyber attacks can be characterized. A comparable approach would also be useful for the generation and labeling process of datasets to better address the intended audiences or to extend existing datasets to include multiple domains.

In our opinion, it is not beneficial to exclude specific attacks or certain classes of techniques. For example, reconnaissance can be performed in many ways. The authors of Maynard et al. [10] and Rodofile et al. [21] define explicit requirements for attack derivation targeting testbeds in this regard. A downside of this approach is that individual requirements explicitly refer only to SCADA systems. A collection of malicious interactions as a benchmark for standardized attacks might be a sensible option for representing specific classes of attacks in existing datasets. Scans, MITM, and C2 configurations could provide a simple and useful baseline for the development of new detection algorithms.

In-depth analysis of real-life security incidents by Makrakis et al. [14], using a superset of the presented incident reports, shows a similar outcome. The authors use a different vulnerability classification, contrary to existing definitions such as by Howard and Longstaff [22], and define five specific vulnerability categories: zero-days, known vulnerabilities (CVE exists), insecure protocols (design vulnerability), insecure configuration (configuration vulnerability), and social engineering. A comparison of documented incidents provides a general classification of the types of vulnerabilities observed in real-world settings. In this case, typical zero-day attacks account for 11% (4/36) of vulnerabilities, which is the smallest portion. A high percentage of vulnerabilities are attributable to known weaknesses (6/36) 17%, insecure protocols (7/36) 19%, configuration weaknesses (11/36) 30%, and social engineering (8/36) 22%. If we attribute insecure protocols to account for existing datasets, we can see a very limited coverage of other vulnerability types (see Table IV). Given that social engineering-type attacks will typically emerge from PERA Layers 4 and 5, exploitation and lateral movement at higher levels are missing in the published datasets. This mismatch between threat landscape and dataset coverage should therefore be addressed in the future.

We suggest that the resulting situation and the trend toward increasingly interconnected systems and sophisticated attacks should be replicated in modern testbeds and datasets. The lower levels of the PERA layers are valuable, but representing the complexities of transitions and interconnections at the upper levels requires extensive modeling, documentation, and assessment for high-quality datasets. Numerous studies explore specific cyber-physical threat scenarios and mitigation techniques for different types of attacks and vulnerabilities. However, there are few testbeds and datasets available to accurately reproduce the complexities and attack sophistication required to represent the full picture (e. g., cyber-to-cyber interactions). In the next sections, we will be discussing how we selected our design criteria based on these findings and how we created our testbed prototype.

IV. Research Context

The idea of a containerized testbed was developed within the context of a research project that focuses on improving the cyber resilience of critical infrastructure communications systems. Key elements include the development of new active and passive cybersecurity techniques for industrial environments powered by artificial intelligence. We conducted

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

literature research to find a suitable testbed that we could recreate. However, this turned out to be difficult due to lack of information or accessibility. As a result, we decided to develop our own testbed to have total control over the entire environment while launching realistic attacks. During this process, we first noticed a shortage of literature that addresses testbed design in industrial environments. Most publications touch upon it briefly, citing a physical system or a reference model as inspiration, as shown in [23]. As a result, we decided to document our process throughout the different development phases.

A. Testbed Design Process

The first task was to choose a common industrial protocol with known vulnerabilities, working exploits, and readily accessible toolkits and Software Development Kits (SDK). Ultimately, OPC Unified Architecture (OPC UA) was selected as the primary protocol for a number of reasons: first, it is well documented in the open standard IEC 62541 [24] and comes with a plethora of security-relevant features, such as encryption, authentication, and certificate management. Second, it supports the creation of complex hierarchical system architectures, which are high-value targets for adversaries due to their highly interconnected monitoring and control infrastructure. Third, there is existing groundwork on published vulnerabilities and exploits for different implementations of production-ready SDKs. Finally, OPC UA is widely used in both industry and research. Due to the large number of existing implementations of the protocol and the extensive configuration capabilities, flexibility would play an important role during ongoing testbed design. Secondary protocols for simulating other legitimate applications in an ICS environment and additional background noise are planned to be included in the future, but are not a primary focus.

OPC UA provides a framework in which complex hierarchical information can be modeled and accessed with standardized services [25][26]. The most basic building blocks are called *nodes*, which are used to construct more complex structures, such as hierarchical data types and objects. The entirety of all node instances is called the *address space*, which defines a standard way of structuring the nodes within in a tree-like fashion. This achieves a consistent way for servers to present data to clients. Initially, we focused on developing prototypes of virtual devices for a custom water treatment testbed with different OPC UA software libraries and versions to reproduce documented vulnerabilities. For all these variants, we manually created all required nodes within the address space. Although we achieved near-identical behavior, the resulting tree structure was not the same, making quick one-to-one replacements of devices during our penetration tests unfeasible. Furthermore, maintaining the different software versions written in their respective programming languages created significant technical debt. Fortunately, OPC UA also allows the address space and custom extensions to be modeled using XML. Through the use of tools such as the *OPC UA Model Compiler*, source code and binary models for specific implementations can be generated using a common modeling language independent of language requirements. Most available stacks support these extensions. Thus, we can use a standard way of building and maintaining devices by utilizing XML-based models and configurations. This simplified the workflow and allowed us to verify a variety of vulnerabilities.

Developing interchangeable implementations was the initial phase in designing a configurable framework. The first concept considered was container technology, which enables packaging software to be built and implemented separately from the underlying hardware. Containers allow deployment on embedded systems with built-in support for different platforms, such as x86 and ARM64. They also allow for more reproducible results, as there is less room for errors regarding software versions, installed drivers, and different toolchains. Replacing containers simplifies exchanging implementations and allows passing additional metadata by using Open Container Initiative (OCI) compliant containers. Other advantages include implicit versioning through tags, high shareability with the research community, and, to a certain extent, the ability to recreate testbeds and verify results independently. In addition, many network simulation tools, such as GNS3, natively support container integration, which facilitates the creation of complex architectures. Finally, the flexibility to replace any container through a real component makes it easy to expand from a virtual to a hybrid setup.

The ability to grow into a hybrid setup proves to be relevant, as commercial, proprietary products historically use reference implementations as a foundation. In 2021, OTORIO [27] released their latest research on OPC UA attack surface, mapping out supply chain dependencies for major manufacturers. Based on the specification available from the standard body (IEC 62541 [24]), there have been different releases of the OPC UA Core Stack for public use. Previous stacks that are no longer officially supported include .NET legacy, ANSI C legacy, and Java legacy. As shown by OTORIO, there is a notable relationship between reference stack implementations and selected OPC UA SDKs. The foundation reference implementations and core stacks have been partly used to design or build commercial and open-source SDKs for products by different OEMs [27]. Due to the ability to include such products in a hybrid setup, we can recreate and evaluate vulnerabilities in these proprietary stacks.

Finally, configuration and bootstrapping issues are separate aspects that we encountered during development. While the underlying protocol has been verified to be securely designed and audited by several bodies (including BSI [28], Kaspersky [29], and Claroty [30]), proper configuration, bootstrapping, and personnel training are still major issues in the industry [31]–[33]. Furthermore, certain implementations are characterized by incomplete feature sets and potentially confusing documentation. We therefore began to customize not just the software stack itself but also the configurations.

Considering these issues with the aforementioned lack of testbed design knowledge, we implement the penetration testing-focused methodology for deriving testbeds proposed

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

by Kraust et al. [23] as a proof of concept. It introduces an iterative, protocol-agnostic approach that gradually builds up a complex testbed from individual devices. During each iteration, penetration tests, including their respective target objectives, are defined and executed. The following iterations build on experience, which allows the user to create more complex attacks over time. In order to achieve this, the testbed must be modular by design. As a consequence, we analyzed the OPC UA protocol in terms of its features and capabilities, which allowed us to extract the basic building blocks of a testbed centered around this protocol. We then used the so-called *Model Compiler* to translate XML files with the specified nodes of OPC UA devices into actual source code reusable across different implementations. In this way, we were able to create applications with the same interface that use distinct software stacks. These building blocks are currently available on GitHub [34][35]. We will explain the full extent of the available software in the following sections.

B. Technical Challenges: Combining HIDS, NIDS, and Agentic Systems

Our testbed addresses the specific challenges encountered during our ongoing research into Operational Technology (OT) security. This research focuses on three main subjects: network-based intrusion detection, host-based intrusion detection, and automated agent-based penetration testing within OT environments. Therefore, the testbed is used as an experimental environment and must meet certain requirements regarding tool integration, the execution environment for agent behavior analysis, and high-fidelity data capture within simulated OT/SCADA targets, which we will discuss in more detail in Section V.

To capture different attack scenarios and operational profiles, we designed a sample application that integrates data generation directly into the testbed workflow (see Section VII). This meant that all data-generating processes had to be considered when designing the capture framework for the respective tasks. While network and host data could theoretically be collected and evaluated in isolation, such an approach diminishes the utility of the dataset and the capture process. We see the most value in the potential to test complex attacks and attempts by the pentesting agent in our environment while simultaneously capturing data generated from the host device and the network. This way, we can provide datasets for academic use, evaluate the impact of various attacks, and assess the effectiveness of our Intrusion Detection Systems (IDS) solutions. This approach is beneficial in the verification of our solutions in a virtualized environment, but can also be used as a reference for designing complex interactions for subsequent IDS research.

A critical challenge in data acquisition inside the testbed is preventing contamination of the dataset through unwanted interactions. Manual intervention or automated setups triggering events, like attacks or operational profiles, will leave measurable traces in the network captures and usually require additional agents to be present on the host systems. As part of the HIDS data-generating process, we retrieve cumulative data directly from the CPU's Performance Monitoring Units (PMUs). Because of this, we cannot run any additional computationally intensive services or (pentesting) agents in the background of a client we want to monitor, as data recording is directly related to usage of CPU and CPU peripherals. Unlike other data sources, PMU readings are difficult to correct post hoc. Therefore, the initial recording must be as pristine as possible. Although errors in the data can still be detected to a certain extent, in our opinion, it is not realistically feasible to correct the data or use synthetic data in this context. To reliably detect client-side attacks and manipulation attempts, it is therefore important to record the data as undistorted and accurate as possible. External interventions over the network are problematic, as they pollute the networked data with potential control traffic and potentially alter the PMU recordings due to the presence of agent software. Contamination also needs to be considered for NIDS datasets and network-based data-generating systems. Background traffic being used to orchestrate tests and control testbed execution can leak into the dataset, leading to over-optimistic evaluation results and a complete failure to generalize to operational environments in IDS research. Similarly, when using agentic systems to monitor or control testbed execution, open or unsecured endpoints can get targeted during automated pentesting operations, contaminating captured data or the testbed configuration itself.

To mitigate these issues, we adopted a scheduled and event-based approach for measurements and interactions, which is discussed in Section VII in more detail. In this context, "interactions" refer to simulated standard traffic between nodes, scheduled configuration changes, and targeted attacks on specific clients.

C. Testbed and Design Requirements

Current security testbed literature covers a number of requirements regarding overall testbed design, dataset creation, and credibility factors related to the intended use case or target domains. To provide a more profound understanding of our current design, we want to present additional requirements and discuss how they compare to our design. Our primary research objective from the beginning was identifying and addressing open issues for testbeds and IDS datasets. After our initial work on existing testbeds we found OPC UA to be underrepresented in existing testbed literature and published datasets and thus focused on building a small-scale setup for security experimentation to allow for dataset creation in the future.

Our initial objective for our testbed prototype was not fidelity but modularity to deliver a more iterative pentesting-based workflow. We wanted to build a setup that allows different, but architecturally similar, testbed configurations to be built on virtualized platforms so that we could explore varying attack surfaces on our simulated system. This provides the benefit of allowing us to compare different attack vectors on known vulnerabilities while capturing datasets of different operational states. In this way, we intended not to record a sin-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

gle long-term dataset but rather multiple short-term operational interactions for our IDS-related research containing state-of-the-art attacks. We therefore opted for our iterative approach, as discussed in Kraust et al. [23]. While usability and cost-effectiveness were concerns, our primary considerations were **modularity**, **openness**, and **flexibility**. In addition to openness and interoperability, we want to discuss the topic of future-proofing with regard to the required simulation technologies and backends used.

Given our available literature, there have been different takes on how to design and document the process of developing a testbed depending on the intended use case. Probably the most comprehensive work on security-focused testbeds currently is the survey by Conti et al. [13] covering six high-level design considerations and general best practices: scope identification, fidelity, expensiveness, reproducibility and comparability, missing representation, and standardization. While generally a quality design aid, these guidelines focus on general concepts and the required steps needed to select or build a full-scale system for a given task. Maynard et al. [10] offer less abstract design criteria: reproducibility, scalability, domain fidelity, process simulation, network emulation support, physical network support, physical device support, and multi-protocol support. Here, requirements are used in a more hands-on approach to design virtualized testbed components and an additional orchestration layer. Ani et al. [36] go into more detail about the design characteristics by evaluating credibility-supporting factors for delivering trust and reliability as an integral part of available research data. They present five overall credibility factors and 14 core characteristics, which are then broken down into different design steps (simulation, architecture, and evaluation). Their credibility factors are (i) design objective and scope, (ii) simulation approach and abstractions, (iii) architecture components, (iv) reflection of core characteristics, and (v) testbed evaluation methods. This work covers high-level design considerations, more hands-on criteria, and different core characteristics that need to be considered for specific use cases. In particular, this work also covers the available definitions of the observed core characteristics used in the relevant literature and how these relate to each other. From a design perspective, these can be used to create precise design goals and design considerations when building a testbed. We will mainly refer to the requirements and definitions introduced by [36] with regard to our design considerations, unless noted otherwise.

Fidelity - Fidelity is typically the most important requirement when building a testbed to estimate changes in the cyber-physical domain or to analyze specific cyber-physical attacks. It dictates how accurately real-world behavior is represented within a customized environment, inherently requiring design trade-offs. A scaled-down version of a physical process can be used to retain high fidelity. A simulation can be used to introduce a less accurate but more flexible process, retaining medium to high fidelity, depending on the type of simulation. A fully abstracted system can be used in case fidelity is less important or the design process requires it. A typical example of this would be a nuclear or chemical process, which can pose real danger in case safety systems are affected during testing. Consequently, testing beyond-safety situations or extreme load experiments often requires isolation through abstraction. Therefore, design requirements dictate to what extent fidelity can be achieved. In our case, physical or process fidelity is not a prime target, since our main goal is to build an open, flexible testbed centered on experimenting with modern protocols and emerging attack vectors. For this reason, we chose to model a highly abstracted water treatment process with the intention of generating representative, moderate-volume traffic for our industrial components, allowing us to test attacks across various interconnected targets.

To our understanding, fidelity covers communication and cyber-physical components. For a network-based system to test and reproduce the exact response of a real system, the use of similar software and system configurations is required. In some cases, even specialized hardware is needed for time-sensitive applications. With our provided container images, we simulate real-world applications by integrating official open-source implementations when testing industrial protocol stacks. However, this covers only the high-level communication aspects of a control system which are required for network-based anomaly detection systems. In the case of cyber-physical-focused systems, feedback between physical (simulated or real) and digital components is of interest. To operate systems within stable and safe boundaries, the type of control (open or closed loop), timing requirements, and distribution of control equipment must be considered. Testbed configurations can range from simple process controls to distributed control loops to operate a system within given parameters. In addition, timing requirements are another important aspect to consider when building a high-fidelity testbed. When performing attacks or destructive interactions, the underlying process requires time to react and time to stabilize during and after testing. If this is of interest, the process (simulation), the control systems, and the communication equipment need to be able to reflect this behavior to generate data suitable for anomaly detection.

Our current design has two main consequences for the provided dataset and the testbed in the long term. The first consequence is the limitation to specific IDS use cases. When building IDS systems to detect nuances or systematic failures in operational process data, getting process timing, process feedback, and correlation of process nodes correct is of utmost importance. Our IDS dataset currently depends on high-level network interactions to determine misuse and manipulation at the SCADA or operations layer, as well as host-specific data at the kernel level or user space. With our current design, the precise timing behavior of the system before, during, and after an attack cannot be accurately modeled at the process level. Therefore, our dataset cannot be used to build reliable detection models for this purpose. Second, as we discussed in Section III, recent changes in malware behavior and malware deployment need to be reflected when performing dataset generating tasks. In our case, we focus on the network com-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

munication aspects, as our data generation process captures data inline between communicating clients and from inside live hosts simulating real-world applications. The latter would not be possible using off-the-shelf industrial devices, as access to hardware-specific monitoring units is typically not supported on industrial hardware, trading flexibility, and openness for fidelity in this case. This allows us to integrate modern threats into our dataset (see Section VIII), which can serve as a reference to other works requiring data more closely related to the cyber-physical domain.

Using complex configurations to build a high-fidelity testbed requires resources, domain knowledge, and long-term planning, resulting in rigid systems that cannot quickly adapt to emerging improvements in security architectures. Our system is purpose-built to capture different operational aspects from the network and hosts, add mitigation strategies, introduce different system architectures, and allow for verification of datasets by post-processing available data sources within the testbed. In this way, we can include modern distributed attacks and simulate Advanced Persistent Threat (APT) behavior in our testbed. We are able to recreate real-world communication profiles because we implement and integrate production-grade industrial communication stacks as part of our testbed prototype. Therefore, attacks targeting specific application stacks or device types can be recreated representatively while lacking fidelity at the process level.

Modularity and *Flexibility* - We implement these aspects at the device and application level of our testbed components. As we will discuss in more detail in Section V, our goal is to implement customized components that allow us to manipulate the network architecture and device-specific configuration by using containerized applications. We use this approach to build heavily abstracted networked components that are arranged to represent different ICS devices for our test cases. At the application level, we provide different software services that are representative of industrial devices, like SSH for configuration, embedded web interfaces for monitoring and updates, and custom interfaces for embedded protocol stacks, like OPC UA in our testbed prototype. This combination of services is used to create an abstraction layer to model industrial devices by combining different components into a single application instance. In this way, we can simulate specific attack vectors by providing preconfigured applications for our testbed. By deliberately relaxing container isolation and enabling resource sharing, we allow security compromises to propagate throughout the entire application. Therefore, enabling us to create opportunities to test lateral movement or command and control capabilities inside our testbed environment, affecting vulnerable services or applications.

Openness - While we are not strictly enforcing openness in the original sense (see [36]) by providing public access to our infrastructure, we base all of our work on open and publicly available tools, frameworks, and technologies. While this is strongly related to repeatability and reproducibility, openness creates additional implications regarding future-proofing and what can be shared across the research community. Our choice of tools and technology starts with the core protocol that we are implementing within the OT environment of our testbed. OPC UA offers publicly available standard documents, tools, references, and implementations. This openness allows us to reuse existing SDKs and build on industry-standard interface definitions (companion specifications) for our testbed components. While we acknowledge the lack of fidelity of our current simulation approach at the process level, we argue that a complex, interconnected system using industry-standard components still offers additional benefit for cybersecurity-related tasks. Using a suitable physical or simulation resource, our current prototype could be expanded to achieve the required level of fidelity by leveraging domain knowledge. However, our attack and data-capturing capabilities could still be used for verification or dataset generation purposes. In the same way, we are capable of sharing not only single component configurations but also prebuilt device images and network architectures by utilizing existing container runtimes. Container development and deployment tools are widely adopted and offer ways that can be used to deploy applications, technology stacks, or device-specific configurations. We would refer to this property of our testbed as *shareability*. The advantages of these tools are that they can be easily integrated and utilized without incurring significant overhead costs as a result of their widespread adoption.

Future Proofing - While often overlooked in prior literature, in our opinion the long-term sustainability of testbeds and simulation backends requires more attention. As discussed in detail by [13] and [36] design objectives and scope must be considered early in the development lifecycle. In cyber-physical setups, the addition of emulation or simulation can be a valuable choice given the required level of fidelity and safe execution needed for a specific use case. However, a consequence of this approach is the introduction of dependencies using aforementioned technologies or highly customized tooling. In practice, the quality of a testbed is bound to the chosen simulation tool or backend. Mininet [12] serves as a prime example; derivative works such as MiniCPS [4] and DHALSIM [11] rely heavily on its specific emulation capabilities. Given the current state of the public repositories, we can observe a lack of maintenance affecting the simulation backend of mininet. While a testbed in itself might not be influenced immediately, missing maintenance in one of its components creates technical debt, hindering usability and compatibility in the long term. We can see similar effects for the testbed published by [10], which uses a custom orchestrator for their deployment of ICS devices. Without proper maintenance, there is limited long-term benefit in using customized tools. Even if these systems are a sensible design choice in terms of fidelity or reproducibility, maintaining these tools comes with an inherent cost. In our case, relying on open-source stacks that provide adequate maintenance and technologies, such as container runtimes that allow easy deployment and customization, should prove beneficial in the long term.

In the previous sections, we have discussed design challenges, presented design requirements, and assessed how fi-

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

delity and future-proofing affect our current testbed prototype. We argue that openness, flexibility, and modularity are key requirements for a modern security-focused testbed to achieve reproducible measurements and reliable data generation for network- and host-based datasets. In the next section we will introduce the current state of our testbed prototype and discuss the current selection of images and configurations required to build a working testbed prototype.

V. MOTRA Research Testbed

The purpose of this section is to introduce the reader to the overall testbed concept and to outline the current testbed design and its individual components. We divide our presentation into four parts: first, we introduce the testbed framework with our core repositories used to build, set up, and operate the testbed in Section V-A. Second, we introduce the current testbed architecture that covers different network domains (Section V-B). Third, we provide a high-level overview that describes our current data generation workflow (Section V-C). Finally, we introduce the general concept of our OT domain prototype using our customizable components (Section V-D). A hands-on configuration and usage example for each of the tools presented is shown in Section VI.

A. MOdular Testbed for Researching Attacks

The MOTRA testbed is built around four different repositories designed to facilitate reproducible industrial security research. The foundation of this architecture is the *MOTRA-Images* repository, which serves as the base layer containing component definitions and reference images to create vulnerable applications or mimic industrial services [35]. By allowing resources to be mixed at the container level, this layer enables researchers to swap underlying implementation types while maintaining consistent application behavior, thereby allowing for targeted vulnerability testing across nearly identical service instances. Given the modular configuration, the testbed components are intended to recreate real or hypothetical system architectures to perform pentesting or data generation. Generally, we do not restrict individual component usage for design purposes. Instead, we encourage the use of container networks to abstract internal logic. By exposing only specific endpoints through the container runtime externally, users can merge various application stacks into highly representative device mockups.

Building upon these base images, the *MOTRA-Setups* repository acts as the configuration layer, defining how these containerized resources are orchestrated into functional devices at the application and architecture level, such as the simulated water treatment processes for the OT enclave (see Section V-D) [34]. The container runtime serves as an abstraction layer, allowing for the separation of internal configuration and simulation logic from externally exposed APIs. This layer leverages Docker Compose to create a single centralized configuration, allowing for the dynamic construction of complex, networked nodes like historians, PLCs, and servers without requiring manual intervention. This approach enables the configuration of specific device parameters and vulnerabilities via a single file, after which the testbed can be isolated from external networks to ensure a controlled testing environment. By utilizing application-specific configurations for multiple devices, networked architectures can be mapped onto real devices using container-capable hardware.

To generate high-fidelity datasets, the testbed utilizes *MOTRA-Capture*, a client-server automation framework designed to schedule and trigger experimental payloads [37]. An important feature of this framework is its ability to terminate the client application during active measurement, ensuring that the target device executes only the measurement software and no other active components. This eliminates network contamination and background noise, producing clean traces essential for host-based and network-based intrusion detection datasets. Finally, the actual offensive capabilities are provided by *MOTRA-Playbook*, a comprehensive collection of attack tools, scripts, and known vulnerabilities packaged into a container environment to allow reuse and verification of our data generation process [38].

Out of the box, the MOTRA framework provides software applications and configuration files: preconfigured Docker container images, Linux-based OS images, sample configurations for containerized networks, and a data-capturing application. To set up a working testbed for a practitioner, additional hardware is required. The current setup offers a fully containerized version for development on a single host using Docker Compose that can be run on common x86- or ARM-based hardware. For a serious setup, networking hardware (routers, switches, and firewalls), capture devices (Ethernet-TAP), and embedded devices supporting container technology are required. Our initial dataset contains setup instructions to configure commodity hardware (Mikrotik Router) to set up a simple DMZ and isolate control zones. In addition to the software components, we also use an integrated simulation process as part of one of our containers, which is explained in more detail in Section V-D. Generally, our component architecture is built to use the container networking layer as an abstraction to hide supplementary services that are mandatory to operate or maintain the physical process of the testbed.

B. General System Architecture

The overall testbed prototype is based on a generic IT-OT system architecture representing different device classes derived from the PERA model. The current version covers a simplified enterprise zone (IT enclave), a control zone (OT enclave), and a Demilitarized Zone (DMZ) connecting both networking zones. The IT and DMZ zones are currently limited to modeling a simplified domain crossover. The entire system architecture of our prototype implementation is shown in Figure 1. Since we currently do not have high-level components that implement a suitable enterprise environment, our current setup only includes a limited number of clients that we are deploying to generate background traffic and optional configuration data. The domain crossover will be utilized in the later stages to incorporate certificate and GDS functionality

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

into our existing testbed so we can evaluate modern security features.

The core architecture of the control zone, covering PERA levels 1-3, was introduced in our previous work on the MOTRA testbed by Kraust et al. [1]. The previous architecture incorporates customizable industrial components, such as intelligent sensors, actuators, a PLC, a historian, a HMI, and a custom SCADA server implementation. The current submission introduces additional components for domain crossover, domain-specific attack tools, and an interaction-driven capture framework for dataset creation (see Section VII). Currently, the only interaction implemented between our IT enclave and our OT enclave is the exchange of configuration parameters and historical data over a shared DMZ database. Our DMZ is built using a single DMZ server and a physical router that spans three different network segments, thus separating IT and OT enclaves. Other works implement network segmentation by integrating separate hardware firewalls [39] or traffic-aware proxies as services [5][7][8]. For our design, we chose a single router with low-level hardware configuration to configure isolated interfaces, inter-domain routing, and firewall rules to keep complexity and cost down.

From a high-level perspective, our system prototype implements two main data flows using different network domains. The core architecture implementing the control zone is built around the centrally located PLC component operating the simulated physical process. Sensors, actuators, and the SCADA server communicate with the PLC, implementing low-complexity process control. To enable data transfer between network domains, the IT components implement an additional OPC UA server and a monitoring server to process historical data and send a subset of control commands to the OT services. The DMZ server acts as a proxy in this setting, as connections can only originate outside the DMZ to the server but not cross-domain. For future improvements, we could expand our current implementation by introducing more complex behavior through software agents, additional services, or background traffic. Existing works [40][41] cover the use of agents to create realistic traffic profiles or the use of bots [8] to forward OT- and IT-related data like planning schedules and historical data between both enclaves. Other options to create artificial data include containerized desktop systems that execute scripts or frameworks that generate random user traffic, which could be used to extend the current prototype in the future.

To perform tests and interactions using the testbed environment, we introduce our automation and capture framework in Section VII. The MOTRA Capture framework requires additional devices to integrate an adversary host as part of the control zone network to perform targeted attacks and generate datasets. Our chosen attack target is the PLC component, as it is the central component of our control zone. A software client is used for automation and the host-side data generation. The additional device (the attacker) will be configured as the capture server to monitor and coordinate attacks and generate network data. For a more in-depth description, see Section VII.

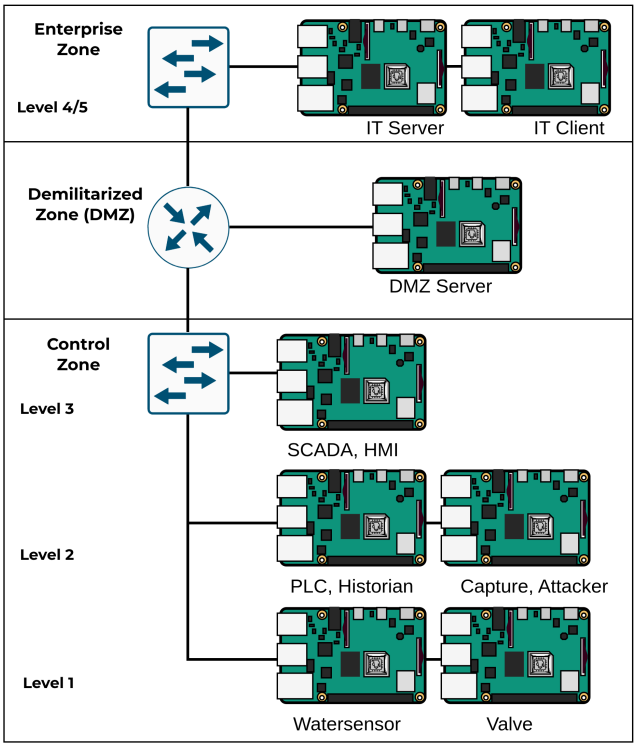


Figure 1. The current system architecture of our testbed prototype covers IT, DMZ, and OT domains. A physical testbed prototype can be built using a open source tool [42] to generate operating system images from scratch using preconfigured setups for our prototype devices.

C. Data Generation Process

The current end-to-end process for data generation consists of five steps, although not all tools for data preparation and export have been made public at the time of submission. To prepare testing activities, we refer the reader to previous work [23], which explains the iterative approach to planning and executing tests within the testbed in theory. In general, we follow the five steps as outlined in Figure 2.

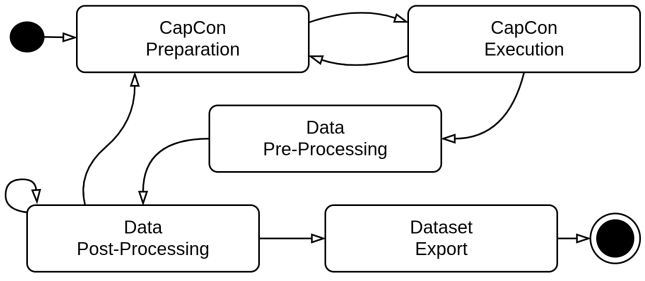


Figure 2. Current end-to-end process for setting up measurements to generate datasets. Using an iterative process throughout capture configuration setup and execution, data is validated semi-automatically and then exported for final release.

During the initial preparation step, capture configurations (CapCon) required for automated test execution are generated. These configurations enable time-controlled interactions to be modeled and executed within the testbed using our

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

capture framework. Live measurement execution is generally performed unsupervised to minimize external influences during data collection. Once the testbed has been successfully configured for test automation (client, interfaces, and network configuration), measurements are processed sequentially, and client data is stored on the capture server. Clients are controlled in between measurements solely by the server to separate configuration traffic from captured process control traffic. The two initial steps (preparation and execution) can be performed iteratively to optimize measurements, add more complex testbed interactions, or introduce different device configurations (e. g., software versions). The third step, data pre-processing, involves the organization of collected data using a database, archiving, and conversion for more advanced processing and to allow for merging of client-side content into a common dataset. Currently, data can only be imported in one transaction, as this process is not designed to be iterative. Data is extracted from the measurement archives of all available clients and linked to existing metadata in the database. In this step, the recorded measurement data is converted into usable data formats (e. g., Parquet) for machine learning and post-processing (e. g., flow data, statistics, and conversations). During post-processing, all available data (network, host, logs) is used to identify anomalies in the individual measurements. This process is currently only partially automated and still requires manual adjustments. This step is required to determine whether destructive tests had an impact on the original testbed configuration and if subsequent measurement cycles failed or succeeded. The final step is to export the collected data from all measurements in a structured format and provide the necessary metadata for a publicly available dataset.

A fundamental problem with current measurements is that the host-side data (PMU measurements) does not provide usable metadata or contextual information and that certain measurements are destructive by design. Based on raw data and the absence of additional context, it is challenging to offer reliable feedback on the quality of the data generated for HIDS. As a solution, we are adopting an approach that automates the data generation process in the testbed and captures a wide range of metadata. Based on the additional data, existing logs, and the PCAPs provided, some fundamental conclusions can be drawn about the measurements. Using an automation harness, we batch process available log data from the testbed and the automation framework with a local LLM instance to determine critical errors, configuration issues, or unexpected anomalies. Additionally, we use network captures from known good interactions to compare different execution cycles based on conversation and flow data to determine if the default configuration of the testbed was altered. By flagging anomalies, we can manually cross-check existing data and repeat failed tasks. This step is intended to make the recording process and the captured data verifiable.

D. Concept and Components

The water treatment process of our testbed is a simplified configuration that is based on critical infrastructure requirements and is implemented as the control zone, as illustrated in Figure 1. The simulated physical process is used as an example of a control process involving different hardware devices with a simulation backend integrated into the container environment as part of the levelsensor component. To integrate various applications and configurations and test individual OT-related vulnerabilities for our research, these components are necessary for our current data generation setup. In the subsequent section, we will introduce the fundamental components that are needed to generate a distributed physical representation of our OT control zone.

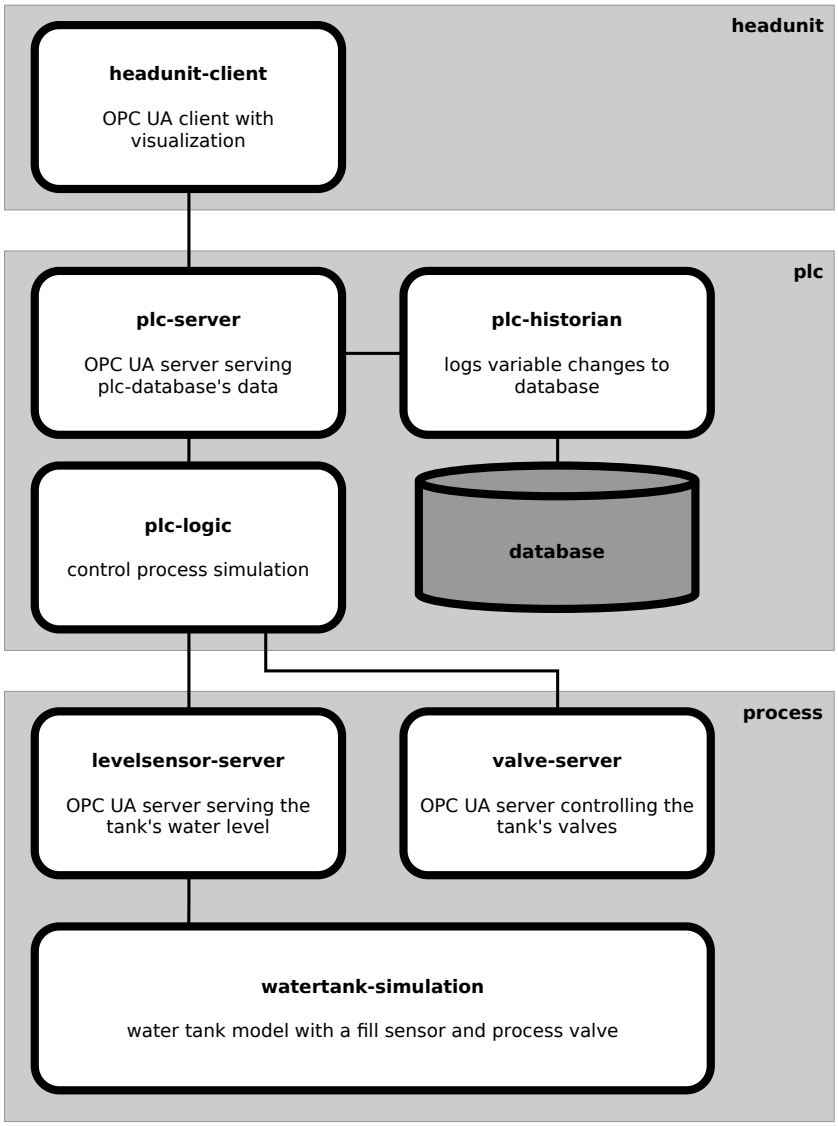
The concept of a flexible testbed through the use of containers allows users to test specific setups and configurations and enables reproducible experiments among cybersecurity practitioners. This requires careful consideration of how to split testbed components and configurations into reusable chunks. During our initial tests, we often found ourselves in the situation of needing another already existing component, such as another sensor or actuator. As a result, packaging software units that perform a certain function was the obvious choice. The exemplary simulated process used in this paper is shown in Figure 3a, where these units (referred to as *components*) are shown as white boxes. For the chosen OPC UA protocol, communication or functional components can be derived in part from the protocol specification, such as client, server, discovery server, or global services.

In addition to the pure functionality, the next most relevant properties for penetration testing considerations are the underlying software stacks and the respective version numbers. Generally, high-level components (e. g., the valve-server) can be assembled using different implementations. Depending on the stack or SDK used, specific vulnerabilities are present and can be exploited. Closely related to this is the selection of the exact software version. As developers constantly patch their software to improve security, many real-world systems do not receive timely updates and continue to run with outdated versions. We account for this issue by allowing for the specification of a certain version as part of the component definition. As a result, we package components according to certain parameters as separate containers (see Section V-D), which is reflected in the suggested naming scheme in the following section. We decided to use Docker as the containerization solution, as it is freely available, well-known, and feature-rich.

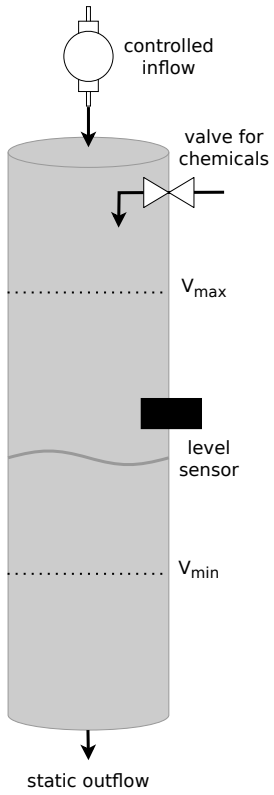
Before explaining the usage and composition of the proposed control zone, we provide a description of the primary dependencies and components shown in Figure 3a. Please note that unless stated otherwise, all components communicate using the OPC UA protocol. In principle, the functionality could be replicated using any other suitable protocol. However, we chose OPC UA since, at the time of publication, there was a lack of available datasets and testbeds that addressed the communication protocol for security purposes.

In essence, the control zone simulates a closed single water tank system as shown in Figure 3b, as an example for critical infrastructures such as water supply. It has a static outflow and an adjustable inflow through a pump. The water level

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org



(a) Schematic representation of the OT enclave architecture. The current configuration features a simulation component interacting with virtual sensors and actuators to establish an abstract physical environment. Process control logic and historical data management are integrated within the PLC component. A head unit client provides HMI capabilities, serving as the central monitoring and control interface for the setup.



(b) Schematic representation of the simulated water tank system

Figure 3. Schematic representations of the OT enclave architecture and the simulated water tank system

is kept between an upper limit (V_{max}) and a lower limit (V_{min}) by activating the pump when the water level falls below V_{min} and deactivating it when it reaches V_{max}. The water level in the tank is monitored by a single fill-level sensor. Depending on the current level, water purification chemicals are added by activating a valve. The control of the valve, logging, and monitoring functionalities are realized by the architecture shown in Figure 3a. In the following, we take a closer look at the individual components.

watertank-simulation - This component implements the simulation process of the physical water vessel and is written in Python. This simplified version is derived from a real process based on a water treatment facility. The current implementation features pump control to keep the water level between the allowed minimum and maximum. The pump is modeled to show PT2 behavior. The current tank fill level is reported to the levelsensor-server via an OPC UA client. For simplicity reasons, there is currently no feedback loop regarding the concentration of water treatment chemicals. The communication between the simulation and the connected

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

servers can be made inaccessible to the host networks by using internal Docker networks while simultaneously decoupling the simulation and the server application code. The realism of the simulations was of lesser concern due to the testbed's focus on penetration testing.

levelsensor-server - This OPC UA server hosts the current sensor readings of the water level sensor. Depending on the requirements of the production network, the security configuration can be adjusted as needed. Another design consideration was the implementation of internal sensor value updates. We went with network-based communication for interacting with the simulation instead of using hard-coded callbacks on each custom server. This allows us to decouple the simulation from the server implementation entirely by linking the simulation backend using container-level networking, which simplifies the replacement of containers and simulation backends.

valve-server - The second OPC UA server allows the control of actuators in the system, which is currently just the valve for adding treatment chemicals. The valve status does not propagate back to the simulation, but we plan to extend the simulation to include this feature in the future. As for levelsensor-server, security features can be enabled as needed.

plc-logic - This component encapsulates the logic of activating and deactivating the valve depending on the current reading of the level sensor. It opens a connection to both the valve-server and the levelsensor-server and subscribes to changes in the water level variable. This triggers the latter to send a message to the logic client, where the reading is first written into a queue and asynchronously evaluated in another thread. Depending on the value, the valve position is changed by writing a new value to the valve-server. Any changes to either the water level or the valve position are also written to the plc-server. This allows other devices (the headunit in our case) to assess process data without having to directly interact and possibly interrupt the process-level servers. Lastly, the water level thresholds for opening and closing the valves used by the logic component can be configured. For this purpose, it is informed if new values are written to the plc-server and adopts the values as soon as possible.

plc-historian - The historian acts as a recording mechanism for all relevant process parameters by writing them into a persistent database. The current implementation uses the file-based, lightweight SQLite database, which is made accessible to the container via volumes. The current database implementation is not packaged as an additional container to allow easier replacement with the user's database of choice and is therefore depicted in gray in Figure 3a. The historian subscribes to all relevant variables on the plc-server and is triggered upon receiving new values.

plc-server - This third OPC UA server allows systems of the upper layers to access process data for monitoring and planning purposes. In contrast to the production network servers, this instance simulates interactions with enterprise clients, e. g., encrypted and authenticated connections for administrative tasks or read-only connections for dashboards. Authorized users may also set certain properties that influence the simulation.

```
services:
  headunit_dashboard:
    container_name: "headunit-dashboard"
    hostname: "headunit-dashboard"
    image: dashboard:latest
    environment:
      - SERVER_URI=opc.tcp://172.17.1.1:4840
    build:
      context: {...}dashboard/python-opcua-
          asyncio/latest
    ports:
      - "8050:8050"
```

Listing 1. Compose specification for an abstract headunit or engineering workstation. Using customized build context, environment, and port configurations, a template image is downloaded, built, and deployed using Docker Compose. After a successful startup, the headunit can be integrated into a physical testbed and executed offline.

headunit-client - This client simulates a control station for visualizing and monitoring the underlying process through a web GUI. It also allows for setting certain process-relevant parameters, such as threshold values. It connects to the plc-server via a secure connection.

Please note that the listed components available at the time of publication are implemented with different software stacks and versions. We implemented only what is currently needed for the specified proof of concept, but it is planned to add more containers. Another notable point is that we will add containers for non-operational network interactions in the future, similar to the authors of [8], to simulate more realistic networks with background noise.

VI. Building and Using Testbed Components

In this section, we present how the previously defined building blocks for the OT domain and supplementary tools can be orchestrated and deployed. As we use Docker, the orchestration tool of choice is Docker Compose [43]. It enables the management of multi-container applications on a single host by using a declarative YAML file. It allows users to define services (components), networks, volumes, and named resources that are then automatically configured and created upon starting the Compose application. Every physical device that is part of the testbed uses its own compose file. This gives the user maximum freedom in terms of distributing services across devices. We intentionally decided against using multi-host orchestration tools, such as Docker Swarm or Kubernetes, as this would introduce additional unwanted traffic that is normally not found in industrial environments.

A. Definition of Architecture and Components

To demonstrate the usage, we again consider the system shown in Figure 3a. For this example, we divided the system into three functional groups: *headunit*, *plc*, and *process*. This indicates a reasonable division of the testbed across different embedded devices, which in our case are Raspberry Pi 4s. The *headunit* could be assumed to be a monitoring workstation, the *plc* replicates the behavior of a real programmable logic

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

```
services:
  # public interface
  plc_server:
    container_name: "plc-server"
    hostname: "plc-server"
    image: node-server:latest
    build:
      context: {...}nodejs-node-opcua/latest
      args:
        NODESET_MODEL: "PLC.NodeSet2.xml"
      {...}
    ports:
      - "4840:4840"
    networks:
      - plc_net

  # internal node
  plc_historian:
    container_name: "plc-historian"
    hostname: "plc-historian"
    image: historian:latest
    environment:
      - SERVER_URI=opc.tcp://plc-server:4840
    volumes:
      - /tmp/docker/database:/database
    build:
      context: {...}historian/python-opcua-
          asyncio/latest
    networks:
      - plc_net
    depends_on:
      - plc_server

  # internal node
  plc_logic:
    container_name: "plc-logic"
    hostname: "plc-logic"
    image: plc-logic:latest
    environment:
      - PS_URI=opc.tcp://plc-server:4840
      - LSS_URI=opc.tcp://172.17.0.1:4840
      - VS_URI=opc.tcp://172.17.0.2:4840
    build:
      context: {...}plc-logic/python-opcua-
          asyncio/latest
    networks:
      - plc_net
    depends_on:
      - plc_server

networks: {...}
```

Listing 2. Compose specification of a multi-service PLC device. This configuration uses a single external node (plc_server) and two internal nodes to implement a historian and the PLC logic control. The NodeSet model provided as build arguments can be used to download and install an information model to reconfigure an OPC UA server dynamically.

controller, and the process encapsulates the interfaces with the low-level process devices. As a result, we would need a total of three compose files (Listings 1 to 3), which are presented below. Please note that the exact usage could deviate as the software matures, so make sure to consult the online documentation for the latest version [34].

This first compose file in Listing 1 configures the services of the *headunit* device. The string for the *build* key points to a valid reference within the MOTRA Images GitHub repository, which is comprised of the 4-tuple *protocol*, *component*, *library/stack*, and *version*. In this instance (Listing 1), the client is implemented using the latest version of the *asyncua* Python library. Non-default configurations for our applications can be configured by using environment variables as part of the service specification (*environment*). In this case, the address of plc_server to another physical device to which the client connects. As this application provides a graphical monitoring interface, it allows access via the host on port 8050 in this example. We use the container runtime to hide control or simulation internals and publish only services intended for the testbed using the ports and network directives.

```
services:
  # public interface with ports enabled
  levelsensor_server:
    container_name: "levelsensor-server"
    hostname: "levelsensor-server"
    image: node-server:latest
    build:
      context: {...}/python-opcua-asyncio/latest
      args:
        NODESET_MODEL: "Tank.NodeSet2.xml"
      {...}
    ports:
      - "4840:4840"
    networks:
      - levelsensor_net

  # internal interface using container networks
  water_tank_simulation:
    container_name: "water-tank-simulation"
    hostname: "water-tank-simulation"
    image: water-tank-simulation:latest
    environment:
      - SERVER_URI=opc.tcp://levelsensor-server
          :4840/KRITIS3M/
    build:
      context: {...}/python-opcua-asyncio/latest
    networks:
      - levelsensor_net
    depends_on:
      - levelsensor_server

networks: {...} # setup dependent
```

Listing 3. Compose specification for an abstract water tank. The configuration creates an internal service using a hidden network (levelsensor_net) and publishes the "real" service using the *ports* directive for testbed-facing applications. This way simulation components can be independently customized.

The PLC application shown in Listing 2 is more complex, as it currently consists of three separate services that communicate over a container local network, *plc_net*. By using these networks, we expose ports only if required. Addressing within the network can be done by referencing the container names using internal DNS resolution. The OPC UA servers in our implementation can be customized by loading a number of different NodeSet configurations, depending on their task. An updated NodeSet will override the default information model of the server and can be used to add a custom device tree (e. g., an additional sensor) to an existing server. This is specified by the *NODESET_MODEL* argument and therefore avoids building the server anew every time the nodes change. By

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

```
tempsensor_server:
  container_name: "tempsensor-server"
  hostname: "tempsensor-server"
  image: node-server:latest
  build:
    context: {...}opcua/server/open62541/latest
    args:
      NODESET_MODEL: "Temp.NodeSet2.xml"
  ports:
    - "4841:4840"
```

Listing 4. Compose specification to extend Listing 3 to include an additional temperature sensor communicating using OPC UA.

```
plc_server:
  # replace specific version
  --- build: "plc-server/node-opcua/latest"
  +++ build: "plc-server/node-opcua/v2.73.0"

plc_server:
  container_name: "plc-server"
  build:
    # replace specific context data
    --- context: {...}nodejs-node-opcua/latest
    +++ context: {...}nodejs-node-opcua/v2.73.0
```

Listing 5. Compose specification to modify an existing image to enable chosen vulnerabilities. Our current reference images are packaged, so modifications to the build context, environment, and build arguments control how components can be used to statically assign known vulnerabilities from within the Compose file.

updating the control logic or adding another internal service, we can use the extended device tree to broadcast additional data to the testbed from behind the container network. Since we are currently using SQLite, we have to mount the database file into the container using *volumes*.

Lastly, the production process application with the simulation is shown in Listing 3. It is demonstrated how internal container networks (in this example: *levelsensor_net*) can be used to separate simulation-specific network traffic from the testbed-facing interfaces of the host system. Therefore, it isolates the simulation and additional tools from the testbed system. It can also be seen that the same server implementation is used, but another NodeSet is loaded upon startup.

B. Iterative Component Modifications

The described exemplary setup can be easily modified. For instance, consider the possibility of introducing an additional sensor to monitor the temperature of the water. To replicate this in a hypothetical testbed, it is sufficient to modify the process compose file by adding a service as shown in Listing 4 to the existing configuration from Listing 3. By deploying the extended configuration on a standalone device, a simplified water-tank simulation using two sensors can be deployed on physical hardware within minutes.

Modularity was a key requirement to support our concept for designing testbeds as proposed in [23]. By starting with a minimal setup initially and gradually adding functionality through additional containers, we can support a bottom-up approach when creating testbeds or experimental environments. This means that penetration tests and measurement deployments are initially performed in a relatively lightweight environment (e. g., only a server-client pair), and the following tests can build upon these layers of understanding. This is more practical than immediately developing complex scenarios.

Another feature of the design is the redistribution of services between physical devices. This can easily be done by moving a service to another compose file and adjusting the environment variables if necessary. This is especially interesting for recording network data at specific nodes. By restructuring the setup, a desired connection can be exposed and recorded by inserting a network TAP.

Lastly, adjusting the setup for a targeted penetration test is straightforward: first, the necessary implementation and version must be selected. This can be done by looking for the right CVEs and selecting the vulnerable software. Next, the image for the affected container is modified in the previously described setup. For example, CVE-2022-21208 describes a denial-of-service attack against implementations using the node-opcua package. Before version 2.74.0, this caused the server to crash if an attacker continuously sent large chunks of data to the server. Simply modifying the version within the compose file enables the vulnerability, as shown in Listing 5.

We currently support the most common open-source implementations of OPC UA, namely open62541 (written in C), node-opcua (NodeJS), opcua-asyncio (Python), locka99/opcua (Rust), and UA-.NETStandard (C#). Over time, we plan to expand on the available stacks and additional vulnerabilities.

C. Image Development and Configuration

Most of the preconfigured reference images can be used for the development and integration of additional services independently. The MOTRA Images [35] repository packages different services, protocols, and extensions that are used by modern industrial devices for integration into an application stack. For ease of development, most images provide a build script and configuration options that can be passed using the existing environment.

```
export CONTAINER_CONTEXT=$(realpath .)
export NODESET_CONTEXT=$(realpath ./nodeset/)
export COMPANIONSPEC_CONTEXT=$(realpath ./cs/)
export CERTIFICATE_STORE=./certs/
./docker-build.sh
./docker-run.sh
```

Listing 6. Setup process for a development image to test different implementations or to prepare a testbed environment.

The example from Listing 6 shows how to run a default OPC UA server standalone. By overriding the context paths, the build process can be instructed to generate a new image using custom device trees or companion specifications. In addition to the server configuration, networking options and server-side security configuration options can be passed globally at this stage.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

```
docker run motra-playbook hydra -L seclists/
    dmiessler-SecLists/Usernames/Names/names.txt
    -P seclists/dmiessler-SecLists/Passwords/
    Common-Credentials/top-passwords-shortlist.
    txt 10.10.100.83 -s 2222 ssh
```

Listing 7. A simple usage example of our tool container. Using the prebuilt container, the available exploits and scanners can be used manually inside the testbed environment or for the automation framework.

D. Adversary Configuration

MOTRA Playbook [38] is a collection of tools, scanners, and security lists for manual or automated pentesting tasks. The tool collection allows building a single container with tools, existing scanners, and wordlists commonly used for performing security tests. We prepared this collection of tools for our automation framework and as a common resource for test planning. The installation scripts for Playbook include references for the setups and image repositories to create a local testing environment. Currently, we include different security lists for brute force attacks, different sniffing tools for reconnaissance attacks, and a list of public exploits and CVEs, which should work for embedded or industrial targets.

E. Data Capture Process

MOTRA Capture [37] is a client-server framework that automates interactions, data capture, and testbed configuration for HIDS and NIDS data generation. For the server, a system-wide installation using systemctl and a systemd unit file is the suggested way of configuration. The client should be installed using the provided pipx installation files. After installation, both client and server require a fresh workspace setup using the provided command-line tools. Once both workspaces are configured and the server is set up with preconfigured capture configurations (see Section VII-D), the testbed automation can be started. To start the server, reloading the system service will update the local capture configuration queue.

```
systemd-run --on-active=30s --unit motra-
    client@initial.service
```

Listing 8. Initial client invocation for the capture framework.

Executing a fully configured testbed configuration is triggered by starting the initial service timer for the MOTRA Capture client. After the assigned timeout, the initial client execution will be triggered, and the measurement process will be performed in unattended mode. During this initial setup time, all active monitoring and configuration connections that are not required for testbed operations can be terminated until all data generation tasks have been completed.

VII. MOTRA Capture - Data Collection and Automation Tools

Datasets are critical for the evaluation of Intrusion Detection Systems (IDS), yet current dataset generation methods suffer from a lack of standardization and reproducibility. As Pinto et al. [44] highlight, the procedures used to generate and evaluate datasets for Network Intrusion Detection System (NIDS) are often highly automated but tend to use different approaches. Specifically, missing standardization for metadata and feature selection is problematic because variations or errors between two datasets can cause issues with different types of models.

Furthermore, the level of technical detail in existing literature varies depending on the domain. In the context of NIDS, emulation-based tools such as Mininet [12], MiniCPS [4], and GRFICS [7] provide a limited level of standardization by using hooks and scripts to capture data at specific interface levels. However, in the domain of HIDS or within our presented literature for OT testbeds, there is currently no systematic process for recording host-based (meta)data. Based on our review of existing testbeds, there is a notable lack of comprehensive datasets featuring detailed logging, Performance Monitoring Unit (PMU) or sensor data.

To address this gap, we introduce a standardized method for recording host data within our testbed environment. Designed around a client-server architecture, our tool can be deployed in a point-to-point configuration to capture hardware performance metrics and log data. Interactions are managed via a network-based interface utilizing a time-based scheduler, keeping network traces clean during live testbed operation. The tool is highly extensible. With minor adaptations, the client can also capture network data, provided that the host is equipped with an additional Test Access Point (TAP) device or out-of-band interface.

A. Connection Protocol and Capture Configurations

Since the term “test” has its own specific meaning in the technical literature and in the field of software testing, we introduce the term “capture configuration” or “CapCon” for short, for data collection and interaction modeling. This term better reflects the intended use case described as part of this work and, more specifically, describes the dependencies with the workloads below. A CapCon can describe one or more workloads that are assigned to one or more clients via a unique ID. CapCons are sent to all active clients for a single measurement or experiment. Each client decides for itself whether the CapCon contains workloads that need to be processed locally by checking the presence of its own ID as part of the workload metadata. CapCons generally serve two purposes: a) to exchange and capture required metadata for each measurement cycle, and b) to serve as a documentation resource to allow for verification of our captured process data. An extension of CapCons is the protocol layer, which will be discussed in more detail in Section VII-D. Capture configurations are built around a standard data validation library to exchange configuration data. The connection protocol is an additional layer built around the CapCons to configure server and client applications to exchange configuration and measurement data.

The term workloads refers to applications, scripts, or executables that are used in the testbed for data collection, attacks, or configuration purposes, to distinguish our configurations

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

from malware payloads and similar terminology. A scheduling procedure is used to determine the exact activation time for the workloads in relation to each other. Start times and timed offsets are used to systematically link actions such as targeted attacks, operational loads, or background interactions in the testbed to individual workloads in order to obtain reproducible results. According to this principle, time-sensitive tasks that need to be customized based on specific events or actions (e. g., timed attacks) can also be modeled.

B. General Concept for Data Acquisition

A systematic procedure was developed to enable the modeling of attacks and interactions in the testbed reproducibly to perform a series of measurements. The general concept uses a client-server architecture that enables the exchange of measurement data and capture configurations via a network connection (web sockets). The general workflow introduces a structured data format that defines a series of arbitrary interactions in the testbed. These interactions are described by a start time, a measurement duration, and time offsets from the start time. A single measurement is defined by one capture configuration (CapCon). Specific workloads are part of a capture configuration and describe how interactions or events are triggered in the testbed. Workloads can be executed by the server and the clients. Based on the time stamp and the metadata of the individual configurations, the interactions in the testbed are documented and reproducible. With our current configuration, the server needs to be permanently active inside the testbed, while all clients can be terminated during an active measurement. Clients are used for instrumenting and automating devices in the testbed and to allow the host capture process to execute.

The general design supports multiple clients, but the current implementation only allows for a single active client. During each measurement cycle, the server will always provide a single CapCon with an arbitrary number of workloads to the client and to the server. Whenever a client is invoked, it will first establish a connection to the server. After successfully connecting, the client archives existing data locally and sends metadata, archives, and results from the last measurement to the server. Next, the client requests a new CapCon from the server and stores it. At the same time, all workloads are extracted, and the workloads to be executed are prepared.

Once a CapCon has been successfully processed, the client sends an acknowledgment to the server. Once all clients are ready and have received the necessary files, an experiment can be started. At this point, the server sends a trigger to all active clients. This trigger defines the start time for all workloads and is used to synchronize the workloads in the testbed. When a client receives the trigger, two actions are performed. First, the Capture Client always schedules the next client activation for the next measurement cycle, decoupling the client from all workloads. Second, all configured workloads are enabled and the required timers are started. After all workloads have been processed, the Capture Client is terminated. This step ensures that no external influences are introduced during the host side measurements by the client.

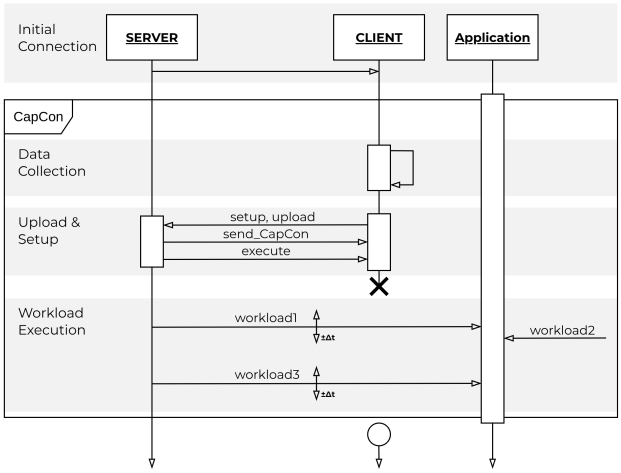


Figure 4. Schematic procedure of an individual CapCon measurement iteration between client and server using three different workloads to test an application.

In the following timeframe, workloads can be used to simulate various measurements, attacker scenarios, or load conditions in the testbed. Once this timeframe has elapsed, the scheduler restarts the client, which first starts a connection to the capture server. The client then checks for available data from the last measurement cycle. During this step, the client packages all available data into an archive and provides any additional metadata that has been requested. Once all archives have been stored locally and the measurement data has been transferred, a new measurement configuration can be requested.

C. Description of Capture Modules

As discussed previously, a client and a server are required to coordinate the individual measurements. To execute and monitor the individual workloads, a scheduler, and an execution module are also required to control any applications, scripts, or services within the testbed. The four essential components of the MOTRA Capture module are briefly discussed in the following.

The server, generally referred to as the capture server, is central to the testbed and has two responsibilities. The first is to control client applications in the testbed. Using the capture protocol described below, multiple capture clients in the testbed can be controlled synchronously, provided that all systems used have a stable time base and are updated regularly. For this use case, the server provides a WebSocket-based API to process requests from clients. The server monitors and tracks the status of the clients during various measurements and ensures that requests and configurations are assigned to the correct clients. In the context of this task, measurement data is also archived locally and collected centrally by the server. The second task involves the server in the testbed assuming the role of an adversary. Since the server has a centrally managed

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

overview of the testbed's status, it is beneficial to monitor and control potentially destructive attacks from a central location. Different types of attacks are performed utilizing the same scheduler and execution module as the capture client.

The client application, also known as the capture client, implements the protocol layer designed for the capture configurations. The detailed procedure for this is described in Chapter VII-D. The client's tasks include archiving and backing up measurement data, transmitting data to the server for processing, extracting workloads, and controlling the scheduler. A state machine is used to control the client, which was also taken into account in the design of the server interface. With the help of the protocol for the CapCons, the state machine is executed by the client and enables synchronized execution of workloads in the testbed. The underlying data validation framework allows only valid workloads and configurations to be transferred and processed. To assign workloads to a client, a unique ID must be assigned during initialization, which can be provided as metadata for the individual workloads. In the future, the ID will also be used to register clients with the server API to manage multiple synchronized clients in the testbed.

The adopted scheduler is part of many Linux distributions and is controlled via *systemd-run*. Systemd offers different types of timers that can either be controlled by providing static configuration files or create so-called transient units dynamically. Individual units (normal or transient) are mapped to timers and can be used to configure applications, control process execution, and monitor resource usage directly from the operating system. A key advantage here is that the interface between systemd and underlying D-Bus can operate with a temporal resolution of one millisecond. With the appropriate configuration, interactions in the testbed can thus be precisely coordinated. Based on the properties of systemd and the possible configurations of the workloads, we used a basic JSON format to define generic structured payloads for scheduling workloads in the testbed. To ensure that the measurements are precisely coordinated, a wrapper for workloads is used as an entry point, which creates a clean environment for each requested attack, measurement, or configuration operation.

The MOTRA Execution Module (MEXEC) serves as the central entry point for all workload executables and is used by both the client and the server. It is implemented as a Python script and initially prepares the runtime environment. It then uses the execvp system call to transparently start the actual target application. The application inherits the preconfigured environment and retains the original Process Identifier (PID) from MEXEC. This offers two significant advantages: first, the constant PID enables precise filtering and tracking of specific applications using available system logs. Second, this design offers excellent flexibility. Almost any native scripts, services, and tools can be integrated into systemd units in this way, configured precisely, and distributed across multiple devices in the testbed without any problems.

The general workflow between Capture Server, Capture Client, and the MEXEC execution module is shown in Figure 5. The MEXEC module is started by activation of a transient systemd unit when a configured timer reaches its activation time. From there, the execution module will parse a local configuration file, identified by its workload ID, and load the environment configuration stored as part of the workload configuration required for the next operation.

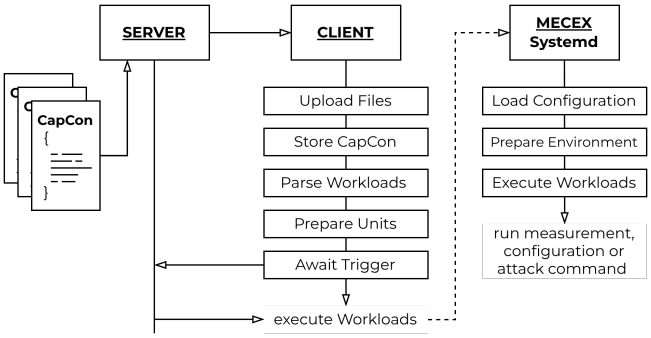


Figure 5. Schematic process flow of the common entry point for client and server applications. Once local workloads are set up, Client and Server will use MEXEC to begin execution at predefined deadlines.

D. Detailed Protocol Design

During the design of our workloads and capture configurations, careful consideration was given to ensure that the architecture for the measurements was reproducible, robust, and extensible. For this reason, a protocol based on Python Pydantic models was designed for the measurement application to build on standardized frameworks. This protocol consists of two parts: the connection protocol and the capture configurations. Both formats are represented as JSON. The connection protocol is an extension of the CapCons and enables the exchange of information between client and server for the orchestration of individual measurements. The capture configurations contain the configuration data required for a single measurement and transfer the workloads that are processed by the execution module.

The general procedure, as shown in Figure 6, involves several steps that must be completed for each client invocation, with a single CapCon measurement configuration processed per request. The following five steps are hard-coded in the client software and help to synchronize the client and server during an active measurement cycle.

- Connection establishment and handshake required for multiple clients.
- Generating metadata and archiving measurement data.
- Data transfer from client to server.
- Request of new CapCon for the next measurement.
- Execution trigger to set the start time for subsequent workloads.

For each individual CapCon, relevant (measurement) data is collected in a local workspace and stored in an archive. The archives are generated and stored locally on the client (step 2) and transferred to the server once (step 3). Subsequently

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

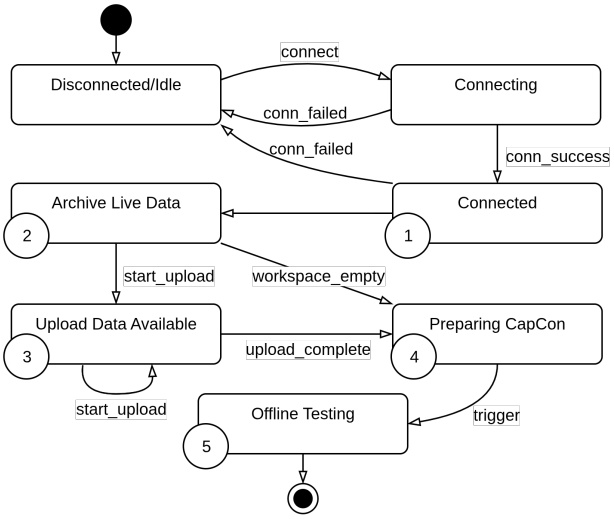


Figure 6. The detailed state machine for the client measurement application. The process shows the entire lifetime of the MOTRA-Capture client. After each iteration, the client is rescheduled for the next cycle and then terminated.

```
{
  "message_type": "CAPCON",
  "timestamp_utc": "",
  "setup": "",
  "cleanup": "",
  "CapConID": "baseline-measurement-0001",
  "description": "A human readable text for the
      measurement",
  "duration": "20s",
  "payload": [...]
}
```

Listing 9. The CapCon header for configuring MOTRA Capture application.

```
{
  "payload_type": "attack",
  "payload_id": "att004-afe4b028",
  "target": ["client"],
  "setup": "",
  "command": "sudo ettercap -T -s 's(230)slqq' -M
       arp  -i end0 /10.10.10.102//
      /10.10.10.103//",
  "teardown": "",
  "description": "MITM attack on PLC",
  "limits": "250s",
  "offset": "500ms",
  "timestamp_utc": ""
},
```

Listing 10. A sample CapCon workload for configuring a MITM attack over a 230s time period using a known conversation target.

(step 4), a new CapCon is requested from the server. A measurement is started by a signal from the server (step 5), which is used to synchronize the start times and to control several clients in the testbed for future extensions. It is important for the measurement process that the clients only become active during the initial configuration step. If the execution trigger initiates a measurement, the scheduler determines the subsequent client execution and terminates the client entirely.

Before each run of an individual measurement, the configurations are stored locally in the workspace on the client and server. This involves the client and server saving the capture configuration and all included workloads to be processed in the workspace as a text file in JSON format. This data not only serves as an aid during testing, but is also recorded as metadata for the data sets and is intended to assist in the subsequent analysis and verification of the measurement results.

The general data format for the measurements consists of one header and an arbitrary number of workloads. The header contains the necessary information related to the next scheduling of the capture client. Each CapCon contains a unique ID (*CapConID*) that is used for identification by the server and later for assigning the workloads. Part of the metadata is a human-readable description (*description*) of the measurement configuration, a timestamp that is filled in when the configuration is processed (when saved), and a *duration* parameter that describes the total duration of a single measurement. The end of the measurement cycle is signaled by sending a capture configuration without CapConID to the client. In this case, the remaining measurement data is recorded, archived, and transferred to the server.

The *message_type* is currently a literal-based parameter intended for handshaking with multiple clients. WebSockets are used to connect to the server, currently with only a single endpoint used for coordination between server and client. Additional message types are planned to control testbed automation and the management of multiple clients. The *duration* parameter is used by the scheduler to determine the start time of the next measurement cycle. This parameter controls the runtime of the individual workloads and the time limit for the entire measurement. Since all processes managed by the scheduler have a limited lifetime, processes that are not completed within the time limit are automatically terminated. Timestamp parameters are part of the header and the workloads. The timestamps are filled in during parsing and storage and are part of the metadata used to verify the individual operations of the measurement using logs and other available data. The *setup* and *cleanup* functions are part of the testbed automation but have not yet been fully implemented.

Individual workloads are represented by a custom data model to trigger interactions within the testbed. The *payload_type* parameter is used to identify actions within the testbed and serves to improve readability and verification. Possible options include *capture*, *attack*, *config*, *logs*, and *other*. The *payload_id* parameter uniquely characterizes a single workload and consists of three elements: the first three letters of the payload type, an incremental count for all workloads of a CapCon, and an eight-digit SHA256 hash that maps the CapCon ID to the workload. Combining the workloads with the hash code derived from the CapCon ID enables the association of log data, metadata, and measurement results following each measurement. The *target* list contains the client-side IDs required to process a workload locally. When

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

parsing the CapCons, the IDs are extracted and compared with the ID configured locally on the client. If one of the target IDs matches the local ID, the workload is assigned to the client, extracted, and processed locally. The *command* parameter is passed internally to the execution module and starts an application or a service. Because it is processed using the *execvp* call, scripts or pipe operations cannot be used internally here. The *offset* and *limits* parameters control the runtime of the configuration. The *offset* parameter controls the time delta between the event trigger and the start of the workload. The *limits* parameter sets the upper limit for the runtime of the interaction for the scheduler. If this limit is exceeded, the scheduler terminates the application for this workload.

E. Preliminary Dataset

As part of our ongoing work, we have published a preliminary dataset that demonstrates the future use of the testbed and the capture framework to produce HIDS and NIDS datasets. The current configuration uses the proposed testbed prototype from Section V-A and provides 155 sample configurations to create baseline profiles and adversary interactions. The reference architecture is implemented in hardware using embedded devices with operating system images built from a companion tool [42]. The custom operating system images were used to integrate our tools and frameworks, set up a static networking configuration, and disable unwanted services and interfaces. Our current device configuration can be replicated using the aforementioned tool, as the core configuration layers are freely available and open source. For our networking deployment, we configured three independent subnetworks: 10.10.10.0/24 for the OT enclave, 10.10.20.0/24 for the IT enclave, and 10.10.30.0/24 for the DMZ services. For the networking setup, physical hardware (MIKROTIK L009UiGS) was used to create a routed layout with a firewall configuration that isolates the IT enclave from the OT enclave.

The current version of the data as well as the detailed configuration for our hardware setup are available on Zenodo [45]. Measurement configurations are hosted as part of the MOTRA Capture framework [37] on GitHub. The supplied Python CapCon module, which can serve as a guide for creating new workloads for customized testbeds, provides a variety of generators to create capture configurations for various use cases.

VIII. Attacks and Measurement Configurations

In order to assess the security of industrial control systems, it is necessary to simulate the threat landscape accurately using virtual testbeds. A key aspect of previous work has therefore been what types of attacks are used in testbeds, how these are parameterized, and why specific attacks were selected for evaluation. To compare existing testbeds, the attack types used must first be defined. For classification, we use the definitions according to Conti et al. [13] for network-based attacks and our previous references. We classify our currently used attacks into seven classes.

Man in the Middle (MITM): With MITM, an unprotected connection can be redirected to a node on the network that is controlled by the attacker. Parts of the packets or entire conversations can be transparently bridged, monitored, and, if necessary, manipulated via the attacker's node.

Injection (INJ): This type of attack describes how invalid or manipulated data can be used to deliberately impact a process or target system through an unsecured channel. External influences can trigger DoS-like conditions or unauthorized command execution on the target system.

Reconnaissance (REC): This generally describes the active or passive acquisition of information within the target network. Device-specific data and conversations are extracted and processed using passive or active methods such as sniffing or probing.

Modification (MOD): This type of attack is often used in the literature in the context of PLCs (programmable logic controllers) and RTUs. It describes a manipulation of the control device in which the running program or firmware (e. g., from the PLC) is modified, deleted, or overwritten through exploitation.

Credential Leak (CL): This type of attack can be related to reconnaissance and describes various methods that can be used to capture specific credentials by observing the network or by examining devices. For example, data can be extracted through sniffing, password cracking, or brute-force attacks. In some cases, however, hard-coded credentials can also be identified through simple device fingerprinting.

Denial-of-Service (DoS): Based on existing literature, DoS is primarily understood to describe overloading attacks. This describes various flooding attacks (DoS and DDoS) with a varying number of attacker nodes, the goal of which is to overload existing resources.

Command and Control (C2): C2 infrastructures use multipart software to overcome network boundaries (e. g., through proxies) and connect isolated nodes into a meshed network. They are used to distribute information, malware, and commands and to maintain the connection to a central server. This decentralized nature of the infected nodes allows commands to be distributed across the OT network, bypassing security features.

The following table provides an overview of established OT testbeds from recent years and the types of attacks they have implemented. This overview serves as a baseline for evaluating which attacks represent the current state-of-the-art.

Command-and-control scenarios are absent from our OT-specific publications, which is a notable limitation. While injection or MITM attacks are frequently depicted in cyber-physical applications, C2 has not been widely considered in the OT sector in recent years (see Table V). We are expanding virtual testbeds through the use of attack vectors and frameworks that are traditionally found in the IT environment. The integration of C2 infrastructures into our testbed (MOTRA) closes the gap between IT and OT security research and enables the simulation of cross-network attacks.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE V. COMPARISON OF VIRTUAL OT TESTBEDS AND IMPLEMENTED ATTACK CATEGORIES. USED TYPES OF ATTACKS: MAN-IN-THE-MIDDLE (MITM), INJECTION (INJ), RECONNAISSANCE (REC), MODIFICATION (MOD), CREDENTIAL LEAK (CL), DENIAL-OF-SERVICE (DOS), COMMAND AND CONTROL (C2).

Testbed	Year	MITM	INJ	REC	MOD	CL	DoS	C2
minicps [4]	2015	•						
VTET [9]	2018		•		•			
Maynard [10]	2018	•	•	•				
GRFICS [7]	2018		•	•	•	•		
RICS-el [8]	2019							
DHALSIM [11]	2020	•			•			
KYPO4INDUSTRY [3]	2020							
EmuFlex [6]	2024	•				•		
LabShock [5]	2025	•	•	•				
MOTRA [1]	2025	•	•	•		•	•	•

While we include attacks and frameworks from the IT domain, we deliberately refrain from attacks in the firmware/program modification category. The reason for this lies in the architecture of our testbed: since our core system is based on OPC UA, we do not use classic programmable control software such as OpenPLC. The pure manipulation of information models does not correspond to classic PLC modification. In order to carry out a comparable attack, modifications would have to be made at the operating system (OS) or firmware level, which is out of the scope of our current work.

A. Baseline Measurements with PERF and TCPDUMP

The baseline for isolated attacks is formed by the data recorded from various capture configurations using our software modules. The objective is to record interactions between different nodes in the testbed and to trigger specific interactions. For our application, we therefore use a node in the network that acts as an adversary and can trigger recordings and attacks on neighboring nodes.

As a baseline measurement, we currently record two different processes within the network. On the one hand, we record the network traffic from a selected node (PLC), as this is where most of the data and connections in the OT network are processed. On the other hand, we record various data points on the host that we want to use for our host-side anomaly detection.

The network-side recordings are created by the adversary node using tcpdump. For this purpose, we have installed an Ethernet tap in the testbed, which records the data transparently and forwards it to the node. This is because it allows us to monitor a specific point in the network without any further configuration in the testbed. Since the adversary node, as the central server, has all the information about a test run at its disposal, we can specifically control when attacks occur in the testbed and when we need to record interactions unfiltered. In this context, baseline measurement means that we can record benign traffic from our water treatment system and use this data as a basis or ground truth for later anomaly detection.

Regarding host-side data, we are currently recording a series of available PMU configurations so that we can select the specific parameters that are relevant for anomaly detection in subsequent measurements. For the measured values, we use the hardware-side monitoring counters, which enable us to monitor selected CPU or peripheral events. The baseline measurement provides us with a basis for anomaly detection, as we will be carrying out various attacks in the testbed and recording the PMU data. Based on these measurements, we then want to evaluate which monitoring units are suitable for specific attacks or attack patterns.

B. Reconnaissance

Various types of reconnaissance operations can be used for scanning networks or infrastructure. Passive attacks are frequently found in the literature, but as part of OT-specific activities, the objective is often to locate critical data or critical nodes in the network. As shown in Section III, the number of documented reconnaissance configurations tends to be rather low compared to other types of attacks. The authors of [14] found that modern incidents tend to rely on off-the-shelf tools for reconnaissance. Some studies also focus more on injection and cyber-physical aspects, which require collection of data during and before the attacks.

In our setup, we show a series of different scans based on the ‘nmap‘ tool with default and scripting configurations to evaluate the effects of scans with varying degrees of aggressiveness. Relevant parameters for the scans include the temporal behavior of the scans, but also the effects of certain scripts and discovery options on the respective devices in the testbed. For example, scripting options can be used to identify vulnerable or unpatched devices. With the different options, we want to analyze how scanners affect devices and how aggressive scans can affect the measurement data of devices on the host side.

C. Injection Attacks Using OPC UA Exploits

The currently available injection exploits for OPC UA have mainly been discovered and implemented by Claroty [30], but no such tests could be found in common testbeds to our knowledge. We therefore used the existing framework and modified certain tests with Scapy to incorporate this class of injection attacks into the data set. The attacks on the available server instances correspond to DoS attacks that consume available resources or connection pools. We use the attacks to measure the impact of exploitation on vulnerable and non-vulnerable services in the testbed for our OT applications.

Since we ported different applications and core stacks when designing the testbed, we can equip individual elements of the testbed with vulnerable implementations and test them accordingly. This allows us to measure significant changes in the performance monitoring units (PMU) on the selected hosts and analyze which attacks have a particularly noticeable effect on the OPC UA server implementations. One major concern with these attacks is that the rest of the system can remain operational to a certain extent and can still respond to certain requests or connection attempts even when under attack. In more complex configurations, the remaining memory gets limited, forcing the system to abort execution mid-processing or fulfill requests slowly.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

D. Man-in-the-Middle Attacks

Man-in-the-middle attacks are already known from previous work on existing testbeds and are typically well understood (see Section III). As part of the literature we have reviewed, sixty percent already incorporate the application of MITM attacks in different configurations. This application has been standard for OT-specific systems to date. We currently implement two different types of attack using the Ettercap tool: an ARP cache poisoning attack and a port stealing configuration. The first type of attack targets a control device in the field, while the other type of attack targets the available networking infrastructure.

Our primary use case for man-in-the-middle attacks is to use the recorded network data as another component of our dataset for network intrusion detection. The recorded data provides a direct relationship between the state of the network and the host. Depending on the type of attack used, the effects of different security mechanisms in the network and on the host can be examined.

E. Flooding Attacks

The subject of DoS-type attacks, like flooding, is repeatedly addressed in OT-specific literature, as certain low-power devices or obsolete systems are susceptible to network overload. To simulate impact on our services, we include a flooding configuration for completeness.

The application of PMUs for these types of attacks is interesting, as the specific effects on modern computing devices with different types of counters and events can be investigated. With proper configuration of our monitoring units, we have the option of monitoring individual processes, CPUs, or peripherals. For example, processes that interact with the network could be grouped together on individual CPUs and monitored collectively. Here, the specific effects of overload on individual system components could be quantified and evaluated.

F. Credential Leaking Attacks

There are different types of credential-leaking techniques, but brute-force attacks are a common example used in work related to IoT and IIoT systems. From the overview in Table IV, we know that testbeds that focus purely on OT or cyber-physical systems rarely utilize IT-specific attacks, as this type of threat is not within the scope of the respective work. However, as we have also seen from existing malware campaigns (Table III), attacks from IT networks are, in fact, a very relevant threat and should be considered accordingly in the design of our measurements.

Based on Daniel Miessler's SecLists [46], we have a significant number of known hardcoded credentials found in different systems, particularly in the OT and SCADA domain, which we can represent in our tests. We use these types of attacks to represent interactions with vulnerable or obsolete devices. Another issue that should not be neglected is the integration of additional technologies in the OT sector due to the increasing connectivity of plants and operator systems. Databases or gateway solutions for data management and centralized processing are increasingly being deployed. Since the number of IT systems at the middle levels of the PERA model is increasing, attacks should be adapted to the target systems accordingly.

G. Command and Control

This type of attack has become a more significant problem in recent years, as it enables an attacker to bypass network restrictions. Given recent developments in OT-specific attacks, we can see different types of command and control capabilities even for PLCs or RTUs. Given that this type of threat has seen limited use in the OT sector, we have attempted to integrate existing frameworks for use in our testbed.

For command and control (C2) applications, three different components are generally required: server, client, and implant. The client is usually the main component that an attacker uses to access or manage a C2 network. The reason for this is simple: access to the infrastructure is typically separated from the attacker's own equipment to avoid leaving evidence on the network and to be publicly available for downstream systems. The server is the central point of contact for clients and implants. The goal here is to control and monitor infected systems. Implants generally refers to the malware that is created for specific target systems. Implants have various tasks, including establishing connections to the server, encrypting connections, transferring data or programs, and providing proxy functionality. Proxy functions are particularly relevant in the context of our testbed, as they can be used to bypass network boundaries. Implants are used at this point to transfer data between firewalls via unfiltered ports or using unsecure configurations. Multiple implants can also establish connections with each other to bridge more complex system architectures.

Currently, there are several tools available that can be used for command and control attacks. Among the better-known candidates are Metasploit, Caldera, and Sliver. Other options include Mythoc/Apfell, Havoc, and Covenant, but some of these tools are no longer maintained or are intended for interactive sessions and are not suitable for our capture framework.

We have decided to use Sliver for this work because we intend to use various payloads that can simulate different types of connections and interactions. Sliver uses implants for communication within the target environment. These small gadgets are created for specific operating systems and hardware architectures and serve as an entry point into a system. These implants are used for communication between one or more infected systems and a central server. By incorporating multiple implants into distributed systems, a network of decentralized nodes can be established that distributes data and commands across networks. In this context, implants can encrypt or obfuscate data and extract it from an otherwise isolated network, take over proxy functionality for other nodes, or execute their own instructions and services on target nodes. The distributed target programs therefore serve to circumvent

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

security functions and isolation within a target network and ensure long-term access to the infrastructure.

IX. Conclusion and Future Work

This work proposes a modular and flexible testbed prototype to facilitate faster reconfigurations for penetration testing purposes. First, we present an overview of previous research on virtual OT testbeds, with an emphasis on OT and cybersecurity. Next, we focus on the existing threat landscape and discuss the state of existing datasets. We put our approach into context by providing design considerations, relevant literature, and current challenges. We discuss design and implementation requirements and necessary tradeoffs. Next, we introduce our testbed by defining the core components of the modular design, the overall framework, and the required concepts. We introduce the MOTRA Capture framework as a significant improvement to our most recent design iteration. This framework is designed to generate host-based and network-based datasets for the purpose of conducting research on anomaly and intrusion detection. We show how the ability to quickly change architecture, implementation, and configuration can be used to leverage penetration testing activities. Finally, we discuss available attack configurations and how we designed the preliminary version of the available dataset.

We argue that existing testbeds suffer from technical debt and rely heavily on custom tools for emulation or orchestration. Taking into account the derivative work of the authors of mininet [12] and the adapting threat landscape, we believe that this will negatively affect other testbeds and datasets in the future. Dependencies for modern testbeds are of substantial value considering the added benefits of fidelity in existing works, but should be chosen with care. MOTRA prioritizes modularity and flexibility, trading fidelity for better maintenance and future-proofing when appropriate.

A significant technical improvement is the ability to record clean data by terminating measurement-specific agents during active recording, preventing the testbed itself from introducing unwanted network or host-side pollution. We hope that the introduction of our framework and the provided design will help in creating more readily available benchmarks for generating live malicious or anomalous data and reproducible scenarios for future IDS development.

We intend to expand the number of available components in order to enable the modeling of more complex scenarios. Setups containing modern technologies like Global Discovery Server (GDS) and Certificate Management are planned to allow testing of different security configurations and device integrations in the future. Based on the current experience with available attacks and configurations, we want to extend our currently available dataset to include different representative multi-stage attacks that could be reused across other testbeds.

Acknowledgment

The presented work is part of the research project *KRITIS Scalable Safe and Secure Modules* (KRITIS³M), which is funded by the Project Management Jülich (PtJ) and the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under funding code 03EI6089A.

References

[1] S. Kraust, P. Heller, and J. Mottok, "A Modular and Flexible OPC UA Testbed Prototype for Cybersecurity Research," in *The Nineteenth International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2025*, Barcelona, Spain: IARIA, 2025.

[2] OPC Foundation, "UA Part 1: Overview and Concepts," 2025, Retrieved: May 25, 2026. [Online]. Available: https : / / reference.opcfoundation.org/Core/Part1/v105/docs/.

[3] P. Čeleda, J. Vykopal, V. Švábenský, and K. Slavíček, "KYPO4INDUSTRY: A Testbed for Teaching Cybersecurity of Industrial Control Systems," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, Portland OR USA: ACM, Feb. 26, 2020, pp. 1026–1032, ISBN: 978-1-4503-6793-6. DOI: 10.1145/3328778.3366908.

[4] D. Antonioli and N. O. Tippenhauer, "MiniCPS: A Toolkit for Security Research on CPS Networks," in *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, Denver Colorado USA: ACM, Oct. 16, 2015, pp. 91–100, ISBN: 978-1-4503-3827-1. DOI: 10.1145/2808705.2808715.

[5] Z. Bernhardt, "Zakharb/labshock," Feb. 6, 2026, Retrieved: Feb. 7, 2026. [Online]. Available: https://github.com/zakharb/labshock.

[6] A. Giehl, M. P. Heinl, and V. Embacher, "EmuFlex: A Flexible OT Testbed for Security Experiments with OPC UA," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, Vienna Austria: ACM, Jul. 30, 2024, pp. 1–9. DOI: 10.1145/3664476.3670931.

[7] D. Formby and M. Rad, "Lowering the Barriers to Industrial Control System Security with GRFICS," *ASE @ USENIX Security Symposium*, 2018.

[8] M. Almgren et al., "RICS-el: Building a National Testbed for Research and Training on SCADA Security (Short Paper)," in *Critical Information Infrastructures Security*, E. Luiijf, I. Žutautaitė, and B. M. Hämmerli, Eds., Cham: Springer International Publishing, 2019, pp. 219–225, ISBN: 978-3-030-05849-4. DOI: 10.1007/978-3-030-05849-4_17.

[9] Y. Xie, W. Wang, F. Wang, and R. Chang, "VTET: A Virtual Industrial Control System Testbed for Cyber Security Research," in *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*, Oct. 2018, pp. 1–7. DOI: 10.1109/SSIC.2018.8556732.

[10] P. Maynard, K. McLaughlin, and S. Sezer, "An Open Framework for Deploying Experimental SCADA Testbed Networks," in *ResearchGate*, Dec. 14, 2024. DOI: 10.14236/ewic/ICS2018.11.

[11] A. Murillo, R. Taormina, N. Tippenhauer, and S. Galelli, "Co-Simulating Physical Processes and Network Data for High-Fidelity Cyber-Security Experiments," in *Sixth Annual Industrial Control System Security (ICSS) Workshop*, ser. ICSS 2020, New York, NY, USA: Association for Computing Machinery, Apr. 12, 2021, pp. 13–20, ISBN: 978-1-4503-9002-6. DOI: 10.1145/3442144.3442147.

[12] K. Kaur, J. Singh, and N. Ghumman, "Mininet as Software Defined Networking Testing Platform," in *International Conference on Communication, Computing & Systems*, Aug. 2014.

[13] M. Conti, D. Donadel, and F. Turrin, "A Survey on Industrial Control System Testbeds and Datasets for Security Research," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2248–2294, 2021, ISSN: 1553-877X. DOI: 10 . 1109 / COMST.2021.3094360.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[14] G. M. Makrakis, C. Kolias, G. Kambourakis, C. Rieger, and J. Benjamin, "Industrial and Critical Infrastructure Security: Technical Analysis of Real-Life Security Incidents," *IEEE Access*, vol. 9, pp. 165 295–165 325, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3133348.

[15] DRAGOS Inc., "PIPEDREAM: CHERNOVITE'S EMERGING MALWARE TARGETING INDUSTRIAL CONTROL SYSTEMS," 2022.

[16] N. Govil, A. Agrawal, and N. O. Tippenhauer, "On Ladder Logic Bombs in Industrial Control Systems," Feb. 17, 2017, arXiv: 1702.05241 [cs], Retrieved: Jan. 31, 2026. [Online]. Available: http://arxiv.org/abs/1702.05241, pre-published.

[17] S. Senthivel, S. Dhungana, H. Yoo, I. Ahmed, and V. Roussev, "Denial of Engineering Operations Attacks in Industrial Control Systems," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '18, New York, NY, USA: Association for Computing Machinery, Mar. 2018, pp. 319–329, ISBN: 978-1-4503-5632-9. DOI: 10.1145/3176258.3176319.

[18] S. Kalle, N. Ameen, H. Yoo, and I. Ahmed, "CLIK on PLCs! Attacking Control Logic with Decompilation and Virtual PLC," in *Proceedings 2019 Workshop on Binary Analysis Research*, San Diego, CA: Internet Society, 2019, ISBN: 978-1-891562-58-7. DOI: 10.14722/bar.2019.23074.

[19] J. Klick, S. Lau, D. Marzin, J. Malchow, and V. Roth, *Internet-facing PLCs-A New Back Orifice*, Blackhat USA, 2015.

[20] M. Yampolskiy, P. Horvath, X. D. Koutsoukos, Y. Xue, and J. Sztipanovits, "Taxonomy for description of cross-domain attacks on CPS," in *Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems*, ser. HiCoNS '13, New York, NY, USA: Association for Computing Machinery, Apr. 9, 2013, pp. 135–142, ISBN: 978-1-4503-1961-4. DOI: 10.1145/2461446.2461465.

[21] N. R. Rodofile, K. Radke, and E. Foo, "Framework for SCADA cyber-attack dataset creation," in *Proceedings of the Australasian Computer Science Week Multiconference*, ser. ACSW '17, New York, NY, USA: Association for Computing Machinery, Jan. 31, 2017, pp. 1–10, ISBN: 978-1-4503-4768-6. DOI: 10.1145/3014812.3014883.

[22] John D. Howard and Thomas A Longstaff, "A common language for computer security incidents," Tech. Rep. SAND98-8667, 751004, Oct. 1998, SAND98–8667, 751 004. DOI: 10.2172/751004.

[23] S. Kraust, P. Heller, and J. Mottok, "Concept for designing an ICS testbed from a penetration testing perspective," in *2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, ISSN: 2768-0657, Jun. 2025, pp. 561–568. DOI: 10.1109/EuroSPW67616.2025.00071.

[24] OPC Foundation, "OPC UA Online Reference - Released Specifications," 2025, Retrieved: Sep. 8, 2025. [Online]. Available: https://reference.opcfoundation.org/.

[25] F. Pauker, T. Frühwirth, B. Kittl, and W. Kastner, "A Systematic Approach to OPC UA Information Model Design," *Procedia CIRP*, pp. 321–326, 2016, ISSN: 22128271. DOI: 10.1016/j.procir.2016.11.056.

[26] S. Friedl, C. von Arnim, A. Lechler, and A. Verl, "Generation of OPC UA Companion Specification with Eclipse Modeling Framework," in *2020 16th IEEE International Conference on Factory Communication Systems (WFCS)*, Apr. 2020, pp. 1–7. DOI: 10.1109/WFCS47810.2020.9114448.

[27] E. Jacob, "A Broken Chain: Discovering OPC UA Attack Surface and Exploiting the Supply Chain," Dec. 2021, Retrieved: Sep. 8, 2025. [Online]. Available: https://i.blackhat.com/USA21/Wednesday-Handouts/us-21-A-Broken-Chain-Discovering-OPC-UA-Attack-Surface-And-Exploiting-The-Supply-Chain.pdf.

[28] BSI, "OPC-UA Security Analysis," Bundesamt für Sicherheit in der Informationstechnik, Tech. Rep., 2022.

[29] P. Cheremushkin and S. Temnikov, "OPC UA Security Analysis," Kaspersky Lab, Security Analysis, 2018.

[30] Team82, "Exploring the OPC Attack Surface," 2020, Retrieved: Sep. 8, 2025. [Online]. Available: https://claroty.com/team82/research/white-papers/exploring-the-opc-attack-surface.

[31] A. Erba, A. Müller, and N. O. Tippenhauer, "Security analysis of vendor implementations of the OPC UA protocol for industrial control systems," in *Proceedings of the 4th Workshop on CPS & IoT Security and Privacy*, ser. CPSIoTSec '22, New York, NY, USA: Association for Computing Machinery, Nov. 7, 2022, pp. 1–13, ISBN: 978-1-4503-9876-3. DOI: 10.1145/3560826.3563380.

[32] L. Roepert, M. Dahlmanns, I. Fink, J. Pennekamp, and M. Henze, "Assessing the security of OPC UA deployments," *Proceedings of the 1st ITG Workshop on IT Security*, May 11, 2020. DOI: 10.15496/publikation-41813.

[33] J. Polge, J. Robert, and Y. L. Traon, "Assessing the impact of attacks on OPC-UA applications in the Industry 4.0 era," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Jan. 2019, pp. 1–6. DOI: 10.1109/CCNC.2019.8651671.

[34] P. Heller and S. Kraust, "Motra-Setups," Laboratory for Safe and Secure Systems, Mar. 3, 2026, Retrieved: May 31, 2026. [Online]. Available: https://github.com/Laboratory-for-Safe-and-Secure-Systems/motra-setups.

[35] P. Heller and S. Kraust, "Motra-Images," Laboratory for Safe and Secure Systems, Apr. 17, 2026, Retrieved: May 31, 2026. [Online]. Available: https://github.com/Laboratory-for-Safe-and-Secure-Systems/motra-images.

[36] U. D. Ani, J. M. Watson, B. Green, B. Craggs, and J. Nurse, "Design Considerations for Building Credible Security Testbeds: A Systematic Study of Industrial Control System Use Cases," Nov. 4, 2019, arXiv: 1911.01471 [cs], Retrieved: Dec. 13, 2024. [Online]. Available: http://arxiv.org/abs/1911.01471, pre-published.

[37] P. Heller and S. Kraust, "Motra-Capture," Laboratory for Safe and Secure Systems, Mar. 8, 2026, Retrieved: Mar. 14, 2026. [Online]. Available: https://github.com/Laboratory-for-Safe-and-Secure-Systems/motra-capture.

[38] P. Heller and S. Kraust, "MOTRA-Playbook," Laboratory for Safe and Secure Systems, Mar. 3, 2026, Retrieved: May 31, 2026. [Online]. Available: https://github.com/Laboratory-for-Safe-and-Secure-Systems/MOTRA-Playbook.

[39] R. Candell, D. M. Anand, and K. Stouffer, "A Cybersecurity Testbed for Industrial Control Systems," 2014.

[40] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, Oct. 2019, pp. 1–8. DOI: 10.1109/CCST.2019.8888419.

[41] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, May 2012, ISSN: 0167-4048. DOI: 10.1016/j.cose.2011.12.012.

[42] Peter Heller and S. Kraust, "Motra-Img-Gen," version 1.0, Laboratory for Safe and Secure Systems, Mar. 14, 2026, Retrieved: Mar. 14, 2026. [Online]. Available: https://github.com/Laboratory-for-Safe-and-Secure-Systems/motra-img-gen.

[43] Docker, "Docker compose manual," 2025, Retrieved: Sep. 8, 2025. [Online]. Available: https://docs.docker.com/compose.

[44] D. Pinto, I. Amorim, E. Maia, and I. Praça, "A review on intrusion detection datasets: Tools, processes, and features,"

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Computer Networks, vol. 262, p. 111 177, May 2025, ISSN: 13891286. DOI: 10.1016/j.comnet.2025.111177.

[45] Peter Heller, *MOTRA Testbed: HIDS & NIDS Dataset for Intrusion and Anomaly Detection*, Dataset, Mar. 2026. DOI: 10.5281/zenodo.18433279.

[46] D. Miessler, "Danielmiessler/SecLists," Mar. 12, 2026, Retrieved: Mar. 12, 2026. [Online]. Available: https://github.com/danielmiessler/SecLists.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Optimizing Certificate Validation in OT Environments by Caching Certificate Validation Results

Steffen Fries*, Rainer Falk*, Andreas Guettinger+
Foundational Technologies*; Smart Infrastructure+
Siemens AG, Germany
e-mail: {steffen.fries | rainer.falk | andreas.guettinger}@siemens.com

Abstract—**User authentication based on digital certificates is becoming increasingly common in industrial environments, also known as Operational Technology (OT). Users may be both human users, as well as technical users like devices, applications, or agents. Certificate-based authentication requires that the relying party must validate the certificate before trusting it. This validation refers to the end entity certificate itself, but also to the intermediate certificates along the certificate chain to the common trust anchor. This poses a significant effort on involved devices concerning computing power, memory, and the complexity of the required validation logic. Different approaches already exist to offload certificate validation from specifically constrained devices. This paper reviews the certificate handling in general and specifically investigates existing optimization approaches. To increase overall efficiency even more, this paper proposes an optimization for infrastructures supporting offloading certificate validation.**

Keywords–cybersecurity; credential; digital certificate; public-key infrastructure; certificate validation, device authentication, industrial security; power system automation.

I. INTRODUCTION

User authentication in Operational Technology (OT), including critical infrastructures is increasingly achieved involving certificates and corresponding private keys as authentication and authorization credentials. Users in this context may be human users, but also technical users like devices, applications, services, or agents. An example of a critical infrastructure is power system automation, spanning from centralized power generation up to the management of increasingly deployed Distributed Energy Resources (DER). Here, substations are used to manage parts of the energy grid and enable transmission of electricity at different voltage levels. Further examples for critical infrastructures are industrial automation systems, like factory automation or process automation as well as intelligent traffic systems, which help to control vehicle traffic on streets or in tunnels.

Utilized credentials are typically managed by a so-called Public-key Infrastructure (PKI) following well-defined operational processes involving Identity and Access Management (IAM) to ensure proper authorization during the certificate lifecycle involving enrollment, update, and also revocation. As outlined in [1], even though certificates are issued by Certification Authorities (CA) of a PKI following procedural security requirements and policies, they need to be validated by a relying party before accepting the certificates to establish trusted communication, or before accepting signed information received from a sender.

Certificates may come in different form factors, X.509 [2] being likely the most prominently used format. It is applied in several security protocols to support secure communication between different entities. Most commonly, Transport Layer Security (TLS) version 1.2 [3] is still widely used, version 1.3 [4] is increasingly applied). TLS supports protecting TCP/IP-based communication, while the complementary Datagram Transport Layer Security (DTLS) [5] supports protecting UDP/IP communications. They employ certificates in the handshake for peer authentication and to negotiate security parameters of the intended communication session. Specifically, TLS is used in different industrial environments to protect domain-specific communication protocols. An example from power system automation is to secure IEC 61850 [6], specified in the IEC 62351 series [7]. A further example from the industrial automation domain is OPC-UA [8], which supports TLS as underlying security protocol.

Certificate validation at a relying party as part of the trust establishment for the interaction can be a time and performance consuming process, as it includes the verification of the certificate itself but also the verification along the certification path up to a common trust anchor (root certificate). This effort becomes amplified when Post-Quantum Cryptography (PQC) is used for X.509 certificates (see also [9]). This is caused by the much larger key sizes for public-keys for some of the post-quantum cryptographic algorithms. The increase in size may turn out to be a problem when employed on constrained devices. The certificate validation policies may become complex, e.g., though the use of multiple or hybrid certificates and may even be operator-specific during the transition phase towards PQC if classical and PQC algorithms are used in combination.

This paper provides insight into typically used certificate validation approaches and proposes a novel approach to keep the effort for the validation of the relying party certificate overall as low as possible. This optimization targets operational infrastructures using constrained devices, which may either not have enough processing power or memory to perform the validation locally or devices in environments, or which do not have access to information from other sources required in the validation procedure. It utilizes available technology but provides an enhancement to also limit the burden on the infrastructure.

The remainder of this industrial research paper is structured as follows. Section II introduces infrastructures for

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

certificate handling, certificates in general, and already known approaches for their validation. Section III introduces an optimization concept of certificate validation services, to allow a relying party to determine trustworthiness in the received certificate more efficiently. Section IV provides a deployment example and Section V concludes the paper and gives an outlook towards future work.

II. Related Technology

This section provides an overview of X.509 certificates, their structure and validation options. It also provides insights into the PKI used to manage the lifecycle of certificates. Specifically discussed is the potential offloading of certificate validation tasks (partially or completely) from an end entity to a supporting infrastructure or communication peer to save memory, time, and processing power. Moreover, local caching of revocation information is further considered, as it is already applied in today's communication infrastructures.

A. Public Key Infrastructure

A PKI provides the technology and processes to manage public keys, or more specifically certificates along their lifecycle. Its setup is depicted in an abstract view in Figure 1. It shows the main components of a PKI, as there are:

- **Registration Authority (RA):** performs the interaction with the requesting entity with respect to identification and authentication of the entity, authorizes the creation of a certificate, and provides necessary information to the certification authority (CA). A Local Registration Authority (LRA) may be used to provide a hierarchy or geographically restricted services connected to a RA.
- **Certification Authority (CA):** issues certificates according to the certification policy.
- **Repositories:** may contain a public directory to look up certificates issued to entities. Typically, it provides information about the revocation status of issued certificates.

Lifecycle refers to different phases during the application for and maintenance of certificates, as there are (not all phases are shown in Figure 1):

- **Registration**: an entity registers and applies for the issuance of a certificate. Depending on the type of entity, it may be identified by different means. This task is typically performed by a Registration Authority (RA), which also verifies the authorization for issuing a certificate to the requester, based on the certification policy. For human users, this may involve identification using identity cards. Devices may be identified using manufacturer-provided credentials, e.g., IDevIDs (Initial Device Identifier) [10].
- **Enrollment**: the process of requesting a certificate for a presented identity and public key. Enrollment of certificates is typically done using an automated approach based on an enrollment protocol. As shown in Figure 1, there are different options for protocols available that address different needs. Besides communication, also the key generation has to be considered. Typically, the entity requesting a certificate generates the public/private key pair and ensures that the private key never leaves the entity. There may be cases where this is not possible, like when an entity does not have enough entropy to generate good cryptographic key material. In these cases, centralized key generation may be more appropriate. Note that the central generation of key material is not supported by all stated enrollment protocols.

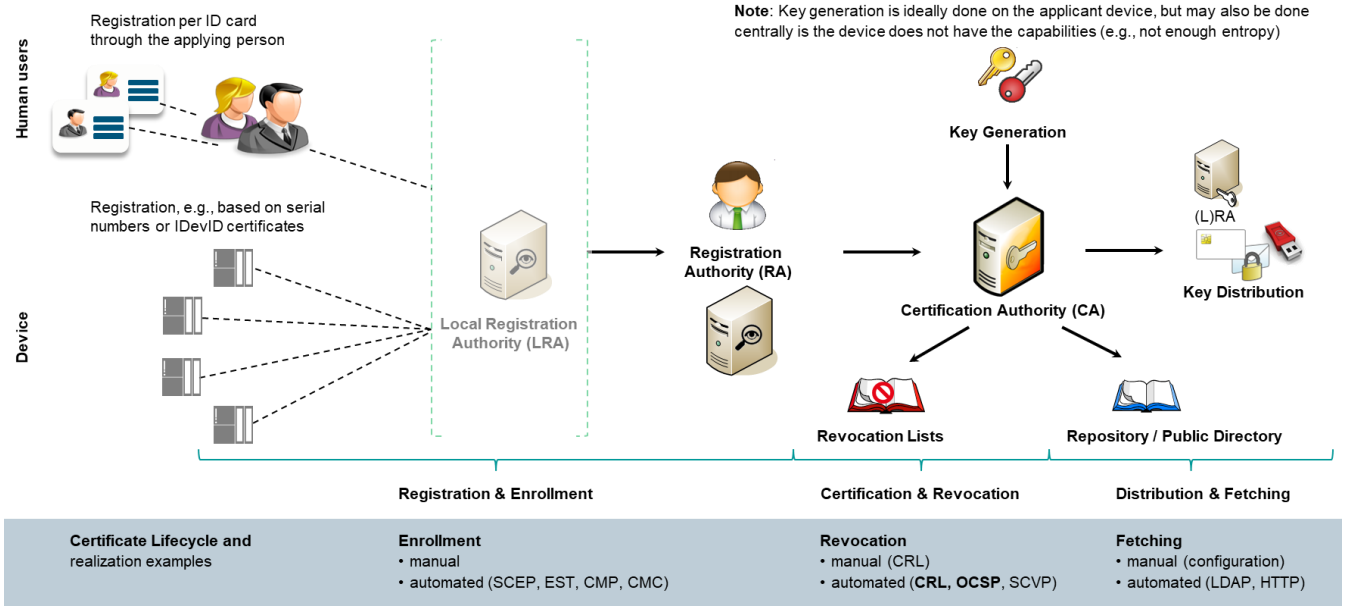


Figure 1. Public Key Infrastructure – General Setup and Interactions

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

- **Certificate issuance:** as part of the enrollment, the CA issues the certificate for the public key presented to the RA. The certificate may contain specific identity information in the `subject` or in the `subjectAltName` component of the certificate. The CA may also include certain restrictions for the contained key in the `keyUsage` or `extendedKeyUsage` extensions. This depends on the certification request, and also on the certification policy of the issuing CA.
- **Certificate update / renewal:** describes the phase when an entity requests to renew the certificate. This may be done by the CA depending on the certification policy. Different approaches are possible, like a simple update of the validity period of the certificate, or the generation of a fresh key pair.
- **Certificate revocation**: is necessary, e.g., if the related key is known to be compromised or if an attribute as a user's department information included in the certificate is not valid anymore, to avoid that the compromised key pair or an invalid attribute information can be used. Depending on an operator's policy, updating a certificate may also lead to the revocation of the old one.
- **Certificate and key archival:** depending on an operator's policy, certificates and corresponding private keys may be archived to enable recovery of lost encryption keys and thus maintain data accessibility.

B. *Public-key Certificates – Structure and Validation*

As stated before, certificates conforming to ITU-T X.509 [2] can be used for different purposes, like entity authentication, e.g., in the context of key establishment in security protocols like TLS or DTLS, or to provide authenticity and integrity-protection of data, e.g., for protecting firmware or software updates.

Figure 2 shows the general concept of a public-key certificate, the binding of the user identity to the corresponding public-key. The user also possesses the corresponding private key, which is kept secret and is used to provide proof-of-possession, which can be verified by the relying party based on the certificate.

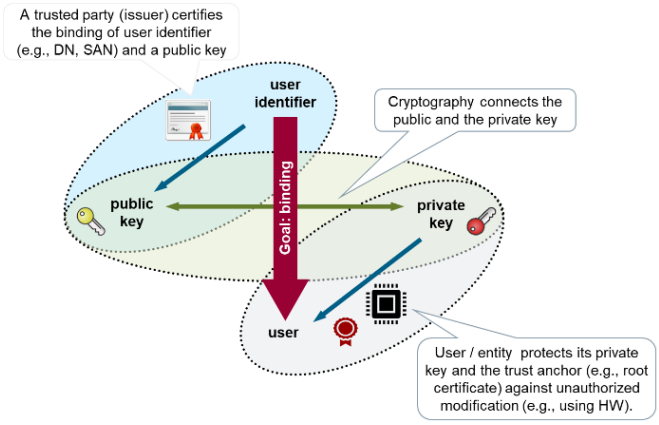


Figure 2. Concept of Binding Public-keys to User Identities [9]

The certificate itself is issued by a trusted third party, a CA of a PKI that digitally signs the certificate during certificate issuing after verifying the user's authorization to get a certificate.

When the certificate is used to authenticate a user, the certificate signature is verified by the relying party as part of certificate validation, similar for all certificates in the path up to a trust anchor (sometimes also called root certificate). Also, further attributes included in the certificate are validated like key usages describing, for which purposes the key confirmed by the certificate may be used.

In addition to public-key certificates, attribute certificates are also defined in X.509. An attribute certificate can be seen as a more short-term enhancement of a public-key certificate. An attribute certificate does not contain a public key, but additional certificate attributes that are connected to the holder of the related public-key certificate [9]. Regarding the certificate validation, which is the focus of this paper, they are handled in a similar way. For simplicity, the paper will therefore concentrate on public-key certificates for the description of the validation optimization, as this is the most broadly used form of X.509 certificates.

```
Certificate ::= SIGNED{TBSCertificate}

TBSCertificate ::= SEQUENCE {
  version                 [0]  Version DEFAULT v1,
  serialNumber                 CertificateSerialNumber,
  signature                    AlgorithmIdentifier{{SupportedAlgorithms}},
  issuer                       Name,
  validity                     Validity,
  subject                      Name,
  subjectPublicKeyInfo         SubjectPublicKeyInfo,
  issuerUniqueIdentifier  [1] IMPLICIT UniqueIdentifier OPTIONAL,
  ...,
  [[2:  -- if present, version shall be v2 or v3
  subjectUniqueIdentifier  [2] IMPLICIT UniqueIdentifier OPTIONAL]],
  [[3:  -- if present, version shall be v2 or v3
  extensions               [3]  Extensions OPTIONAL ]]
  -- If present, version shall be v3]]
} (CONSTRAINED BY { -- shall be DER encoded -- } )
```

Figure 3. Public-key Certificate structure (see [1])

ITU-T X.509 [1] defines the structure and content of public-key certificates, as well as the verification of the components. As shown in Figure 3, the certificate is a structure signed by the CA, containing the `subject` as the name of the entity (user) and the `subjectPublicKeyInfo` structure with further information about cryptographic algorithm and the contained public-key. The signature is created typically using traditional asymmetric cryptographic signature algorithms like Rivest Shamir, Adleman (RSA) or Elliptic Curve Digital Signature Algorithm (ECDSA). Different key sizes are supported by these signature algorithms. As outlined in [9], PQC algorithms are increasingly demanded to address potential threats in the advent of a cryptographically relevant quantum computer. A PQC algorithm considered as replacement is for instance Module-Lattice-Based Digital Signature Algorithm (ML-DSA, formerly known as CRYSTALS-Dilithium [10]), which has a much larger key size compared to traditional cryptographic algorithms.

During the certificate validation, several components of the certificate are verified. Depending on an organization's security policy, the minimum set of components of an X.509 certificate to be verified comprises the

- expected identity (typically contained in the subject or subject alternative name),
- validity period (from/to),

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

- key usages and potentially extended key usages,
- signature of issuing certificate authority.

In addition to the information contained in the certificate directly, the certificate revocation state is checked. This information is commonly provided by the issuing CA and indicates if the certificate has been revoked before the validity end, stated in the certificate, has been reached. Revocation may be done if the certificate or the corresponding key has been compromised, or the certificate was superseded. The revocation information can be fetched from the CA via different approaches like using

- Hypertext Transfer Protocol (HTTP) to query a Certificate Revocation List (CRL, see C.1) below). A CRL is provided by the issuing CA and typically updated once a day.
- Online Certificate Status Protocol (OCSP, see C.1) below) to get most current revocation information.
- Server Certificate Validation Protocol (SCVP, see C.2) below) to offload the complete certificate validation.

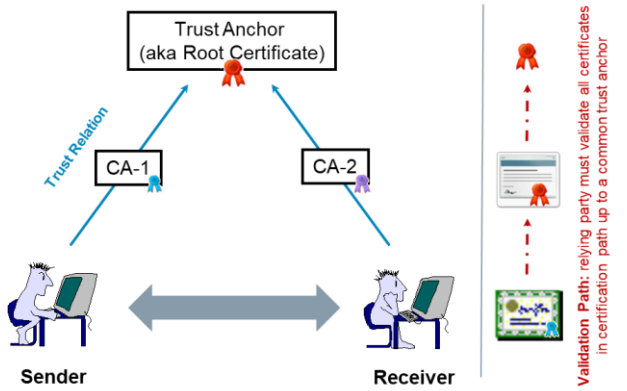


Figure 4. Certificate Path Validation

As stated before and depicted in Figure 4, the verification must be done not only for the end entity certificate, but for all certificates in the certificate chain up to the trust anchor, including the certificate path validation and the verification of the certificate revocation state, which also requires communication with different issuing (intermediate) CAs.

C. Certificate Validation Support Approaches

The following subsections describe potential approaches to support certificate validation and relate to either revocation state information or to certificate components in general.

1) Online Certificate Validation Protocol

CAs typically provide Certificate Revocation Lists (CRLs), containing information about revoked certificates, signed by the CA. These lists may grow and may be difficult to handle, specifically on constrained devices. CRLs are generally distributed by a CRL distribution point to which at least temporary access is necessary. An alternative is the use of the Online Certificate Status Protocol (OCSP, IETF RFC 6960, [12]). It enables clients to query the revocation state of single or set of certificates via an OCSP responder. This lifts the handling of complete CRLs from the clients. OCSP support needs an online connection to the OCSP responder. OCSP responder URL and CRL-DP URL are included in issued certificates. As the CRL, the OCSP response is a signed statement that needs to be validated.

2) Server Certificate Validation Protocol

Applying OCSP, as shown in the previous subsection, still requires validation of certificate components locally on the verifying device. A further approach exists, which delegates the certificate validation to a central authority. It is specified as Server Certificate Validation Protocol (SCVP, IETF RFC 5055, [13]) and allows a client to send the certificate in question and a validation policy to the SCVP server, which takes over end entity certificate validation, certificate path construction, and certificate path validation. This increases efficiency on the client side, but still poses load to the server side, specifically in networks with a high number of clients employing certificates more frequently. The final answer from the SCVP server is also a signed statement. The approach proposed in Section III kicks in here to optimize server-side processing.

3) Certificate Authorization Validation Lists

The complete opposite way to certificate revocation is the explicit authorization of certificates using so called Certificate Authorization Validation Lists (CertAVL, see ITU-T X.509, [2]). They constitute allow lists which explicitly provide information, which certificates are considered trustworthy. This allows to offload revocation handling to the central point creating the CertAVL. X.509 also defines critical extensions to mandate the validation of a CertAVL before accepting it. In contrast to CRLs or OCSP responses, CertAVLs are managed by the system operator, not by the issuing CA. Like CRLs, CertAVLs also have a validity period that is determined based on an operator's security policy and may be also in the range of 24 hours.

4) DNS-based Authentication of Named Entities (DANE)

A further approach is known as Domain Name Service-based Authentication of Named Entities, (DANE, IETF RFC 6698, [14]), which is protected by DNSSEC (IETF RFC 9364, [15]). It enables domain administrators to specify the keys or certificates used by TLS servers as DANE TLSA resource record. The DNS administrator for a domain name is typically authorized to specify identifying information about the zone. Supporting DANE, he also makes an authoritative binding between the domain name and a certificate used by a TLS server in that domain. Thus, a TLS client trusts the certificate information received via DNSSEC, after validation of the DNSSEC signature. It avoids certificate validation, as it got authoritative information from the DNS server.

5) OCSP Stapling

A further approach is known as OCSP stapling, specifically in the context of TLS. Using OSCP stapling a constraint device can request an OCSP response from the remote site and thus avoid separate communication with an OCSP responder. This may be adventurous in situations when the requesting peer has either communication restrictions and may not reach the OCSP responder or if the OCSP communication protocol is not implemented.

For TLSv1.2 [3], this feature is specified in IETF RFC 6066 [16] as a certificate status request (`status_request`)

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

and response extension allowing TLSv1.2 to provide an OSCP response for the server certificate along with its certificate. As this extension only allows to provide a single OCSP response, a further extension is defined in IETF RFC 6961 [17] for multi-stapling, allowing to request (`status_requestv2`) and staple OCSP responses also for intermediate CA certificates contained in the certificate list of the server certificate message. TLSv1.3 [4] provides support for requesting and stapling OCSP responses as described in IETF RFC 6066 for all certificates in the certificate list provided by the client or the server side. As it works in both directions it can accommodate situations in which the server is the constraint device, and the client is more capable peer. An example scenario would be web-based access to communication controllers.

D. Caching of Revocation Information

A common practice to avoid fetching fresh CRLs whenever a certificate is received and validated, is the usage of CRL caching (see also [1]). CRLs contain information about CRL issue time and when the next update will be provided. This allows a local implementation to cache the CRL for the period until the next update. Caching avoids additional communication and decreases the processing time for certificate validation when handling certificates from the same issuing CA. The downside of caching until the next update is that emergency updates during the validity period of the CRL may not be recognized. Caching of revocation information is also the base for the proposed optimization described in Section III.

E. Avoiding Revocation Handling

A fundamental different approach for the certificate revocation check is the usage of short-lived certificates to avoid the need to perform a certificate revocation check.

In such a scenario, or in general when the certificate lifespan is shorter than the time needed to detect, report, and distribute revocation information, the usage of CRLs or even OCSP might be omitted without increasing the risk of a misuse of a potentially invalid certificate. This obviously requires a highly automated and reliable certificate management to ensure that renewed or updated certificates are available at the entities when needed.

Certificates may also be issued with a "`noRevAvail`" extension, as defined in IETF RFC 9608 [18], to indicate that no revocation information is provided at all. Supporting short-lived certificates requires a certificate management system with at least sporadic (e.g., once a day) CA connectivity to enable certificate renewal or update. Note that this connectivity requirement is also given for regular revocation state checks, which are typically done every 24 hours.

III. Proposed Optimization of Certificate Validation Service

As discussed in Section II, different approaches are already available to offload certificate validation from a client. Not all of them are equally suited for OT networks. For instance, DNS is not always available, which limits the possibility to utilize the DANE approach outlined in Section II.C.4) For OT networks, specifically the use of allow lists as in Section II.C.3) or the complete offload of certificate validation as in Section II.C.2) becomes more interesting. While the use of SCVP optimizes the client-side operation, the handling of the SCVP response server can be optimized, too. This is the focus of the novel approach in this section.

As specified in IETF RFC 5055 [13], an SCVP client sends a request containing the certificate to be validated including specific verifications to be done, like the construction and validation of the certification path, key usages, etc. The validation result will be provided to the requesting client, which in turn only needs to verify the server's signed response. To optimize the SCVP server handling, it is proposed that the result of a certificate validation or certificate chain validation is provided on an SCVP Response Collector (SRC in Figure 5).

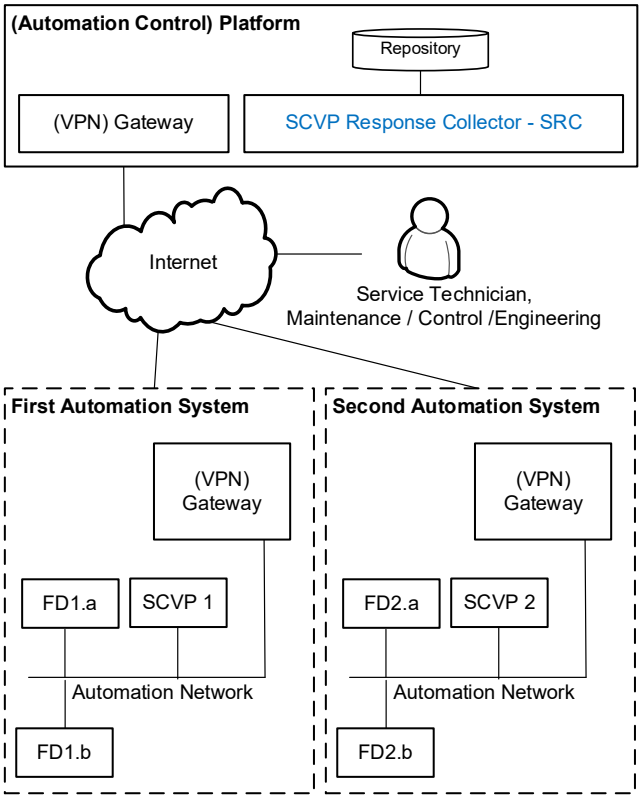


Figure 5. Example Setup for SRC operation

This information can be used to reduce the response time for different types of queries:

- for client queries for the same certificate, e.g., in scenarios where a higher number of clients communicate with one service
- for client queries for the different certificates from the same issuing CA by specifically caching the certificate chain validation.

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Thus, it is no longer necessary to perform all validations separately. The SRC can provide the validation result via different mechanisms, like:

1. publishing the result of the validation of the certificate and/or the certificate chain in a public directory (e.g., accessible via LDAP, HTTP, FTP, ...),
2. publishing the result of the validation of the certificate and/or the certificate chain using in hash chain-based ledger technology (e.g., Ethereum, Hyperledger).

Note that the choice of realization of the SRC specifically for a chosen ledger technology may have an influence on the validation effort. This counts for both the infrastructure for the SRC, but also on the client side for the interaction to query and process a SCVP response.

In addition, the security policy of the organization may need to consider that caching of validation results provides an optimization but also requires further consideration in an organization's security policy. An example is the storage duration of validation results to ensure it matches freshness requirements.

Figure 5 shows two automation environments (e.g., production environment, substation, etc., designated as First and Second Automation System. In each automation environment, an SCVP server (SCVP 1 and 2) works locally as proxy for certificate validation and certificate chain validation. In both automation environments, local communication and communication with outside the domain (e.g., for remote maintenance, access to information in other networks) takes place. As soon as a Field Device (FD) of the automation environment 1 makes a certificate validation request, this is processed by the local SCVP 1 server. If a client (e.g., FD1.a) is allowed to process cached responses, SCVP 1 can first ask the SCVP Response Collector (SRC), which may be public or part of the control center, whether corresponding validation already exists. If not, it carries out the validation and makes the response available to the field device FD. At the same time, it publishes the result of its validation in a repository as a signed data structure (signed with the private key of the SCVP 1) and thus makes the information available to the SRC for subsequent requests.

The connection to a repository can be realized either via LDAP or via ledger technology. The described interaction and abstract message flow is shown in Figure 6.

The described approach addresses the design goal to benefit the delegation of certificate validation on (constrained) clients to a more powerful centralized service, and to optimize the backend service operation.

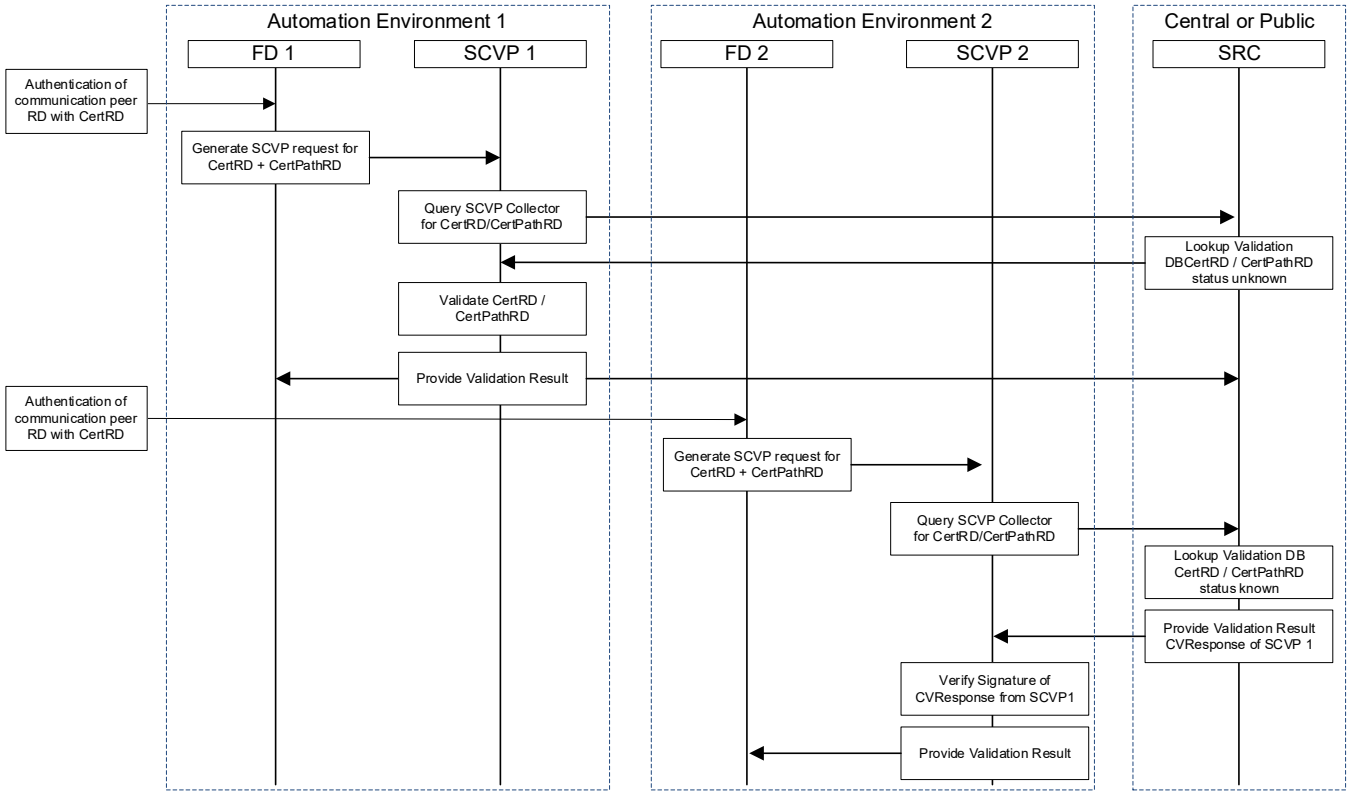


Figure 6. Example Call Flow to utilize SCVP Response Caching

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A locally cached certificate validation result supports the availability of the automation system even if the central validation service should temporarily not be available. In addition, the applied protocol SCVP does not need to be changed as the optimization proposal only addresses the handling on the server side. This ensures that clients already utilizing SCVP can be kept unchanged, still being able to leverage the optimization.

IV. Example Deployment Scenario

The motivation for the proposed optimization is the improved operation of critical infrastructures like complete power systems. As shown in Figure 7, the power system architecture consists of different infrastructures, as there are the electrical infrastructure facilitating generation and transmission of generated power as well as the communication infrastructure to enable control and maintenance of the electrical infrastructure and the connected power flow. Power system automation utilizes the communication infrastructure to control power generation and distribution. Substations are deployed to support connectivity of networks with varying power levels, e.g., to connect generation with the transmission network (high voltage) and further to the distribution network (medium voltage) down to the low voltage network. Substations contain different types of sensors to measure, e.g., voltage, current, phase angle, or the oil temperature of transformers. They also host different actors like protection devices that can trip a dedicated line, e.g., when certain values are out of range, to protect the infrastructure from outages caused by shortcuts or overload. The communication between different parts of the power systems can be done by applying the standard IEC 61850 [6]. IEC 61850 describes not only the communication protocols utilized between devices in a substation, but also the communication with a control center or for remote access. In addition, it specifies the common information of real devices as data model. As stated before, the communication between these devices can be secured by applying IEC 62351 security controls [7] to protect the communication within the substation, but also with other substations, remote maintenance or a control center. IEC 62351 security controls completely rely on certificate-based authentication employed in the key agreement to protect IEC 61850 communication and also related communication connections for engineering or interactions with repository services.

A closer look at the internal network structure and component placement in substations is shown in Figure 8. A substation is typically divided into zones or levels to better separate different security domains. This follows design recommendations specified in the standard IEC 62443 to increase security, and to decrease the attack surface in different administrative domains.

Sensors and actuators are placed on field level and communicate between each other like shown in the lower right side of Figure 8 using Generic Object-Oriented Substation Event (IEC 61850 GOOSE) and Sampled Value (IEC 61850 SV) exchanges between protection devices. In addition, there is communication with station controllers residing in the substation's station level and accumulated data for monitoring and control. Also on station level, a Demilitarized Zone (DMZ) is placed, which provides connectivity to the substation external network, e.g., to connect to a control center for further accumulating monitoring data and providing an option to visualize and control the complete power system of an operator. The DMZ provides additional separation and contains several security relevant components, necessary for operation.

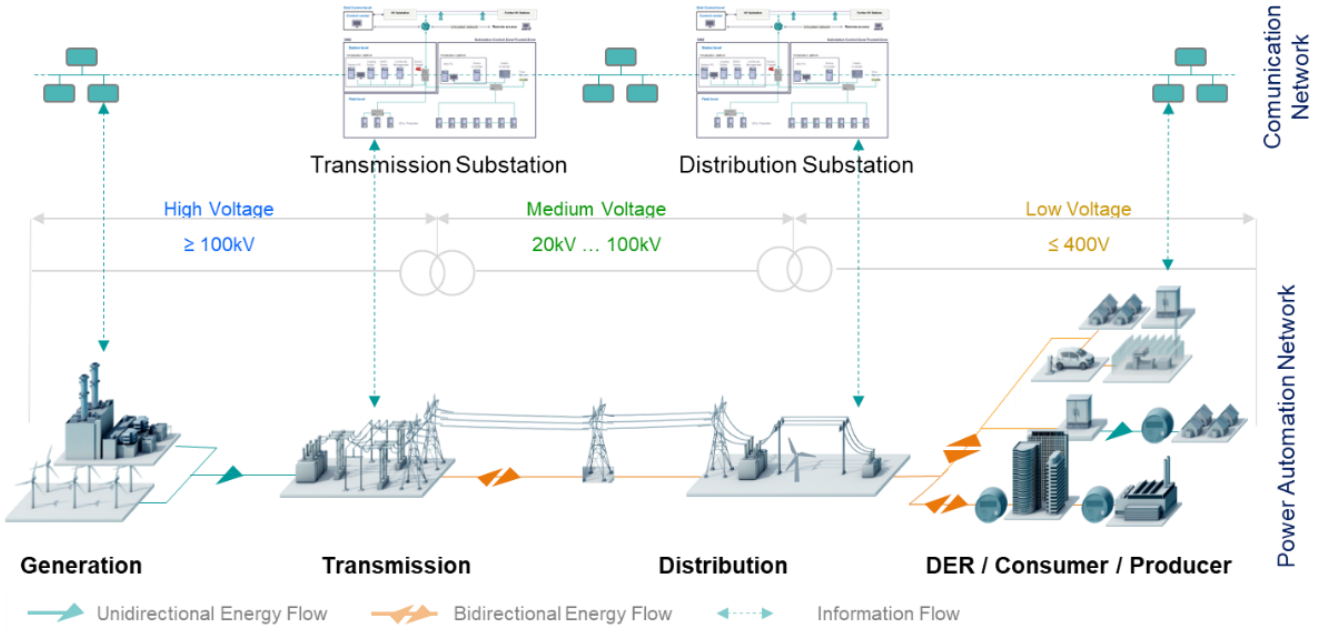


Figure 7. Example deployment scenario: Power system infrastructure

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

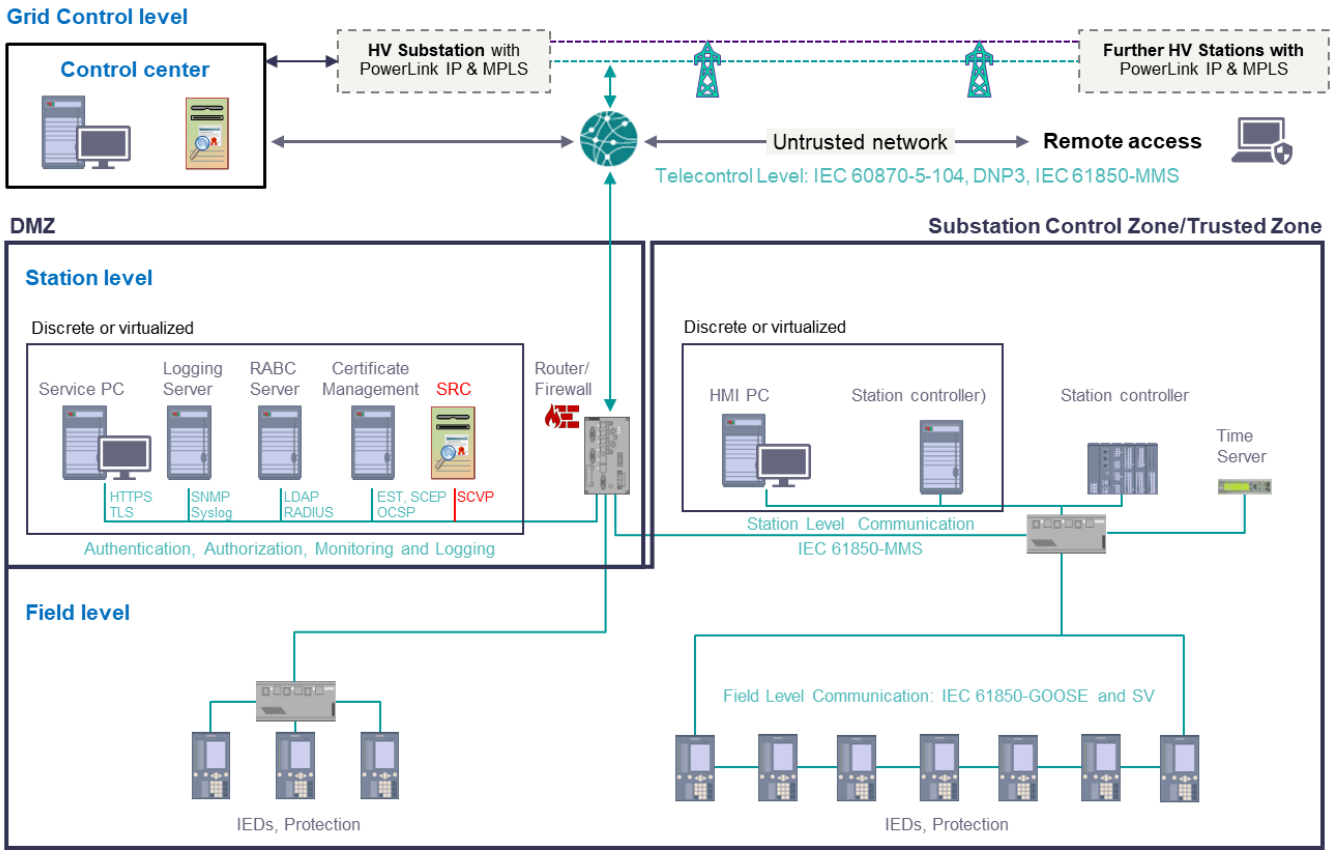


Figure 8. Example deployment scenario: Digital substation network structure

Examples are logging components to allow analysis of potential failure cases. They are specifically important in case of security breaches to allow for forensic analysis of what happened. A further component is a LRA that provides a substation-local interface to PKI services. Note that depending on the actual operational setup, PKI services may be provided directly in the substation or may be hosted in the control center, to which the LRA then interacts.

Like the LRA, the proposed SRC is also a component providing services for different devices within a substation, but potentially also across substations. It can therefore be deployed in different locations of the power system architecture, as there are:

- In the control center to allow an optimized response for the complete network the operator is responsible for. This service may then be utilized by single or multiple substations, depending on the operator's power automation system.
- In the substation's DMZ to ensure availability of information also if connectivity to the control center is interrupted temporarily. An SRC in each substation is required to apply the certificate validation optimization. Note that when using inter-substation communication, certificate validation is likely performed for peer certificates from outside the receiving substation domain. This may require connectivity from the SRC in the receiving domain to the issuing CA in the sending substation to ensure revocation information for the certificate is considered.

The final placement of an SRC is expected to be done based on the operational policy and security policy of the operator and may be done in dependence of the deployment approach of the operator's PKI (centralized in a control center or decentralized in the substations).

V. Conclusion and Future Work

This paper provides an introduction to the use of certificates and certificate validation in OT, focusing on energy automation. It introduces certificate management utilizing a PKI and specifically addresses necessary steps during the certificate validation by a relying party. Focus is placed on an optimization to offload efforts for certificate validation, including the validation of the certification path and revocation state of involved certificates to device external services. The proposed solution simplifies the implementation on constrained devices, e.g., by avoiding additional communication protocol stacks to be supported, and it enhances availability of the automation system if the functionality of a central CRL or OCSP responder in the infrastructure is temporarily not available or connectivity to these remote peers cannot be guaranteed from the OT environment. Furthermore, rather complex policy decisions

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

on certificate validation can be consistently applied for different components of a substation.

The novel approach proposed to optimize the operation of a certificate validation infrastructure in an automaton environment utilizes SCVP as a standardized protocol and combines it with caching of certificate validation information in the backend service. With this approach an OT system, like a power system automation system outlined as example deployment in Section IV, is enhanced with (potentially local) caching functionality for certificate validation information. As caching directly relates to the freshness of validation information, the caching time is a parameter to be considered in an organization's security policy and that has to be set up during the engineering of the system. The caching time will typically be determined based on a risk-based approach and may vary between installations.

At the time being, only the general concept has been developed and has been discussed in the context of a power system automation scenario. The next consequent step is a practical evaluation of the proposed solution regarding the impact to the overall system, based on an implementation to prove its efficiency and effectiveness.

Specific points for the evaluation besides the provisioning of cached certificate validation information may comprise the analysis of the

- Performance impact on the client when using cached validation response instead of local calculation of the validation. This may be considered specifically with different certificate path length.
- Impact on code size for the client side as communication protocol stacks for selected protocols like OCSP or CRL fetching may be omitted.
- Impact on the infrastructure site, e.g., handling additional components, depending on the chosen approach for the publishing of validation results as outlined in Section III.

While the described approach focuses on complete offloading certificate validation comparisons may also be done with partial offloading of revocation checks by using OCSP or by neglecting revocation checks completely by relying on short-lived certificates. Note that the latter still requires to verify revocation states of issuing CAs, as they typically do not use short-lived certificates.

REFERENCES

[1] S. Fries, R. Falk, and A. Guettinger, "Optimizing Certificate Validation in OT Environments", IARIA International Conference on Emerging Security Information, Systems and Technologies SECURWARE 2025, October 2025, [Online]. Available from https://www.thinkmind.org/library/SECURWARE/SECURWARE_2025/securware_2025_1_10_30012.html, [retrieved May 2026]

[2] ITU-T X.509 ISO/IEC 9594-8:2020, „ITU-T X.509 Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks", 2019, [Online]. Available from: https://www.itu.int/rec/T-REC-X.509-201910-I/en, [retrieved: May 2026]

[3] T. Dierks and E. Rescorla, IETF RFC 5246, "Transport Layer Security (TLS) Protocol v1.2", August 2008, [Online]. Available from https://tools.ietf.org/html/rfc5246, [retrieved: May 2026]

[4] E. Rescorla, IETF RFC 8446, "Transport Layer Security (TLS) Protocol v1.3", August 2018, [Online]. Available from https://tools.ietf.org/html/rfc8446, [retrieved: May 2026]

[5] E. Rescorla, H. Tschofenig, and N. Modadugu, IETF RFC 9147, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", April 2022, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc9147, [retrieved: May 2026]

[6] IEC 61850-x, "Power systems management and associated information exchange", [Online]. Available from: https://webstore.iec.ch/en/publication/6028, [retrieved: May 2026]

[7] IEC 62351-x, "Power systems management and associated information exchange – Data and communication security", [Online]. Available from: https://webstore.iec.ch/en/publication/6912, [retrieved: May 2026]

[8] OPC-UA, "Open Platform Communications Unified Architecture", [Online]. Available from https://reference.opcfoundation.org/, [retrieved: May 2026]

[9] S. Fries and R. Falk, "Supporting Cryptographic Algorithm Agility with Attribute Certificates", IARIA International Journal of Advances in Security, 2024 vol 17 no. 1&2, pg. 92-98. [Online], Available from https://www.iariajournals.org/security/sec_v17_n12_2024_paged.pdf, [retrieved: May 2026]

[10] IEEE 802.1AR, "IEEE Standard for Local and metropolitan area networks–Secure Device Identity", August 2018, [Online]. Available from https://ieeexplore.ieee.org/document/8423794, [retrieved: May 2026]

[11] L. Ducas et al., "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme", 2017, [Online]. Available from https://eprint.iacr.org/2017/633.pdf, [retrieved: May 2026]

[12] S. Santesson et al., IETF RFC 6960, "X.509 PKI Online Certificate Status Protocol - OCSP", June 2013, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc6960, [retrieved: May 2026]

[13] T. Freeman, R. Housley, A. Malpani, D. Cooper, and W. Polk, IETF RFC 5055, "Server-Based Certificate Validation Protocol (SCVP)", December 2007, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc5055, [retrieved: May 2026]

[14] P. Hoffman and J. Schlyter, IETF RFC 6698, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", August 2012, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc6698, [retrieved: May 2026]

[15] P. Hoffmann, IETF RFC 9364, "DNS Security Extensions (DNSSEC)", February 2023, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc9364, [retrieved August 2025]

[16] D. Eastlake 3rd, IETF RFC 6066, "Transport Layer Security (TLS) Extensions: Extension Definitions", May 2011, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc6066, [retrieved: May 2026]

[17] Y. Petterson, IETF RFC 6961, "The Transport Layer Security (TLS) Multiple Certificate Status Request Extension", June 2013, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc6961, [retrieved: May 2026]

[18] R. Housley, T. Okubo, and J. Mandel, IETF RFC 9608, "No Revocation Available for X.509 Public Key Certificates", June 2024, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc9608, [retrieved: May 2026]

2026, © Copyright by authors, Published under agreement with IARIA - www.iaria.org